

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL
REPUBLICII MOLDOVA
IP CENTRUL DE EXCELENȚĂ ÎN INFORMATICĂ ȘI
TEHNOLOGII INFORMAȚIONALE
CATEDRA INFORMATICĂ II

ANDRIAN DASCAL

PROGRAMARE STRUCTURATĂ ÎN C++

*Îndrumar pentru activitățile practice,
destinat cadrelor didactice și elevilor din IP CEITI*



CHIȘINĂU, 2021

Elaborat conform Curriculumului modular „Programarea structurată”, reactualizat în 2020. Discutat și examinat la ședința catedrei „Informatica II” din 5.11.2021, Proces Verbal Nr. 3, șef catedră, Luminița Gribineț

Autor:

Andrian Dascal, magistrul în Informatică și Tehnologii Informaționale, grad didactic II, IP CEITI.

Recenzie:

Galina Luncașu, prof. la discipline de informatică, grad didactic superior, IP CEITI.

Redactare științifică:

Ioan Jeleascov, magistrul în Informatică și Tehnologii Informaționale, consilier în proiectarea arhitecturii software în cadrul unității "StoneHard" din Londra, Marea Britanie.

Toate drepturile asupra acestei ediții aparțin autorului. Orice tipărire sau retipărire fără acordul în scris al autorului atrage răspunderea potrivit legii.

© Andrian Dascal, 2021

DRAGI PRIETENI,

„Programatorul este un creator de universuri pentru care doar el este responsabil. Universuri de complexitate practic nelimitată pot fi create sub forma programelor de calculator. ”

Joseph Weizenbaum

Programarea structurată este o paradigmă a programării informatice apărută după anul 1970 datorită complicării crescânde a programelor de calculatoare. A apărut ca un model nou de programare, în scopul de a crea noi tehnici de programare apte de a produce programe care să fie sigure în funcționare, pe o durată mai lungă.

La originea programării structurate se află un articol publicat de E. W. Dijkstra în anul 1968 și intitulat „Go To Considered Harmful”, în care se arată că, pentru a reduce complexitatea programelor și a ușura înțelegerea acestora, este preferabil să se elimine din programe instrucțiunea GOTO (mergi la) prin care, în limbajele de programare procedurale, se indică saltul de la o instrucțiune la alta. Ulterior, Dijkstra împreună cu alți cercetători în domeniul informaticii au conceput o teorie coerentă a programării structurate, conform căreia la conceperea unui program, este recomandabil să se respecte următoarele principii: (1) teorema de structură: orice program poate fi întocmit folosind numai trei structuri de control fundamentale: structura secvențială, structura alternativă și structura repetitivă; (2) la conceperea programelor se recomandă să se aplice tehnica de elaborare descendentă, numită și tehnica rafinărilor succesive; (3) domeniile de valabilitate (de vizibilitate) ale variabilelor și structurilor de date trebuie să fie limitate.

Prezenta lucrare reprezintă un suport praxiologic având drept scop însușirea de către elevi a cunoștințelor necesare gândirii algoritmice și formării culturii informaționale prin prisma rezolvării problemelor, incluzând peste 100 de probleme model rezolvate însoțite de comentarii corespunzătoare.

Acest îndrumar este destinat cadrelor didactice debutante și elevilor, dar și fiecărui om cult ce va activa într-un mediu bazat pe cele mai moderne tehnologii informaționale. Mult succes la fiecare și realizări frumoase !

Autorul

CUPRINSUL

INTRODUCERE	5
1. MEDII DE PROGRAMARE PENTRU LIMBAJUL C/C++	
1.1 Prezentarea mediului de programare CodeBlocks 20	10
1.2 Prezentarea unui IDE pentru smartphone	16
1.3 Prezentarea unui IDE pentru Linux	18
2. APLICAȚII FUNDAMENTALE	
2.1 Aplicarea transformărilor elementare	21
2.2 Aplicarea formulelor matematice	27
2.3 Aplicarea operatorului condițional	33
2.4 Probleme propuse	38
3. APLICAȚII ALE INSTRUCȚIUNILOR DECIZIONALE	
3.1 Aplicarea instrucțiunii IF	45
3.2 Aplicarea instrucțiunii SWITCH	51
3.3 Probleme propuse	58
4. APLICAȚII ALE INSTRUCȚIUNILOR CICLICE	
4.1 Aplicarea instrucțiunii FOR	62
4.2 Aplicarea instrucțiunilor WHILE și DO . . . WHILE	67
4.3 Particularități ale instrucțiunilor ciclice	73
4.4 Probleme propuse	79
5. APLICAȚII ALE TIPULUI DE DATE ARRAY	
5.1 Aplicarea array-urilor unidimensionale	85
5.2 Aplicarea array-urilor bidimensionale	92
5.3 Probleme propuse	106
6. APLICAȚII ALE TIPULUI DE DATE STRING	
6.1 Aplicații elementare cu string-uri	115
6.2 Probleme propuse	126
7. STUDIUL INDIVIDUAL GHIDAT DE PROFESOR	
7.1 Lucrarea nr. 1 – Probleme elementare	128
7.2 Lucrarea nr. 2 – Structuri de control	130
7.3 Lucrarea nr. 3 – Array & String	135
7.4 Model de lucrare pentru studiul individual	138
BIBLIOGRAFIE	164
ANEXE	
A1 Operatorii și instrucțiunile limbajului C/C++	165
A2 Funcții din fișiere antet	168
A3 Model de testare finală asistată de calculator	175

INTRODUCERE

Odată cu 2020 care deja s-a sfârșit, iar C++ 20¹ urmează să intre pe piață, acum că standardul C++ 20 a fost finalizat, întrebare: „Ar trebui să învăț C++ în 2021?”, sună ca de bun simț. La urma urmei, actualizarea C++ 20 se spune că este la fel de revoluționară ca și cea C++ 11 care a modernizat limbajul și l-a salvat de la o anumită obsolescență și de la modul Cobol. Învățarea unui nou limbaj de programare, în special unul la fel de complex ca C++, este o investiție semnificativă în timp, așa că știind dacă va fi util, „Va fi util”.

1. IMPORTANȚA STUDIERII LIMBAJULUI C++

Patru adaosuri majore:

1. Conceptele se asigură că datele utilizate într-un șablon îndeplinesc un set specificat de criterii și verifică acest lucru la începutul procesului de compilare. Prin urmare, compilatorul poate furniza un mesaj de eroare scurt și semnificativ dacă nu este îndeplinită cerința definită a unui concept.
2. Corutinele sunt funcții generalizate care pot fi suspendate și reluate păstrându-și starea. Din cauza lor, programarea asincronă în C++ poate deveni mainstream. Corutinele sunt baza pentru sarcini de cooperare, bucle de evenimente, fluxuri de date infinite sau conducte.
3. Intervalele sunt noi iteratori pentru intervale precum vectori sau liste. Aceștia acceptă algoritmi care pot opera direct pe container, pot fi evaluați fără efort și pot fi compuși.
4. Modulele pot fi folosite pentru a împărți blocuri mari de cod în părți logice. Acestea duc la timpi de compilare mai rapizi, fac redundante fișierele de antet și pot ajuta la eliminarea soluțiilor macro.

Tehnologii larg cunoscute

Comunitatea open source Google are peste 2.000 de proiecte scrise în C sau C++. Amazon (comerț web), Facebook (social media), Amadeus (bilete de avion) și Bloomberg (informații financiare) au părți critice scrise în C++. Sistemul de operare Android se bazează pe o versiune modificată a nucleului Linux care este scrisă în C și C++.

1 Standardul C++ 20 și C++ 23: <https://isocpp.org/std/the-standard>

La fel și MySQL, care este cel mai popular software de baze de date open source din lume. Microsoft Office, Internet Explorer și Visual Studio sunt scrise în Visual C++. Companii precum Lockheed Martin, Boeing, Northrop Grumman etc., fac publicitate continuă pentru inginerii de sisteme încorporate cu abilități C++. Nevoia de întreținere, extindere și testare a codului este enormă.

Jocuri, operațiuni intensive destinate procesoarelor vector, oferindu-le o productivitate mai mare. Dezvoltarea jocurilor a fost un domeniu major de aplicații pentru limbajul C++. Limbajul oferă posibilitatea de a suprascrive blocurile de performanță, cum ar fi alocarea memoriei. Are capacitatea de a structura zonele de date și de a gestiona memoria. Mai mult decât atât, permite programarea procedurală pentru operațiuni cu procesor intensiv, făcându-l destul de rapid și oferă un control mai mare asupra hardware-ului. Multe motoare de joc sunt scrise în C++.

Sisteme integrate, de ingineri calificați în C++ pentru dezvoltarea sistemelor încorporate, pentru modularea semnalelor la software-ul lor. C++ se găsește pe scară largă în sistemele încorporate. Există o cerere masivă de ingineri de sisteme încorporate calificați în C++. Domeniile includ software pentru automobile (de exemplu BMW), software de control al zborului (de exemplu Lockheed Martin), controlul rachetelor, motoare marine (de exemplu motoare pentru nave MAN), controlul turbinei eoliene (de exemplu Vesta), tomografie computerizată și instrumente științifice. Multe echipamente medicale avansate, cum ar fi aparatele RMN, folosesc C++ pentru scriptarea software-ului lor. C++ face, de asemenea, parte din aplicațiile de inginerie, cum ar fi sistemele CAD / CAM de ultimă generație.

Predare și cercetare

Limbajul de programare C++ este, de asemenea, utilizat pe scară largă pentru predare. Este o opțiune excelentă pentru predarea conceptelor de bază sau avansate. De exemplu, siguranța tipurilor, gestionarea resurselor și invarianții pot fi analizate cu C++. Cele 3 principii ale programării orientate pe obiecte - încapsulare, moștenire și polimorfism - pot fi adoptate folosind C++, la fel și șabloanele de clase și ierarhiile de clase și modelele de proiectare. Este, de asemenea, minunat pentru a demonstra algoritmi avansați.

Câteva numere sau date importante

Numărul programatorilor care utilizează în prezent C++ este de milioane. Aplicațiile din lumea reală care au fost dezvoltate folosind C++ sunt numărate cu sutele de mii. Liniile de cod sursă C++ implementate în prezent în aplicații numără miliarde. Numerele ar trebui să convingă pe oricine că C++ este unul dintre cele mai utile limbaje de programare de învățat.

C++ este un limbaj de programare orientat obiect care a fost conceput pentru a fi un instrument puternic pentru construirea unei largi varietăți de aplicații software. Dar acesta nu este principalul motiv pentru învățarea C++ în 2021 și nu numai. Există un număr mare de programe de misiune critice scrise în C++ care nu pot fi înlocuite cu un alt limbaj care oferă același nivel de viteză și control asupra resurselor sistemului.

2. IMPORTANȚA ÎNVĂȚĂRII CONȘTIENTE A LIMBAJULUI C++

Principiul învățării active și conștiente cere ca elevii să-și însușească ceva în măsura în care au înțeles și să participe singuri la aflarea cunoștințelor pe care profesorii le propun la un moment dat. Înclinația către activitate este naturală la elev, drept pentru care cadrul didactic va încerca să valorifice atent această preocupare.

A fi activ, înseamnă a gândi, a medita, a raționa, a merge pe firul unui gând sau al unei idei. După cum putem observa, avem de-a face cu o activizare transcendentă, o adâncire și interiorizare subiectivă, o autolămurire sau edificare pe cont propriu. Această stare poate fi atinsă acordând elevilor autonomie și încredere, prin creditarea lor cu realizări maxime, prin folosirea inteligentă a puterii predice în învățare, știind că profesorul este un fel de sculptor transcendent în școală, care făurește comportamente viitoare ale elevilor.

Participarea conștientă alimentează dorința de cunoaștere și de implicare. Nu trebuie încurajată învățarea pe de rost, însușirea mecanică a informațiilor; accentul se va pune pe înțelegerea prealabilă a cunoștințelor înainte ca ele să fie stocate sau reproduse.

Orice cunoaștere presupune receptare și descoperire, iar aceste strategii promovează învățarea conștientă, care este opusă celei bazate pe memoriza-

rea mecanică. Învățarea prin descoperire este complementară cu cea bazată pe receptare și este definitorie prin faptul că elevul „trebuie să regrepeze informația primită pentru a obține (el însuși) generalizarea înainte de încorporarea ei în structura cognitivă”. Astfel, complexitatea învățării prin descoperire reflectă dimensiunea obiectivelor situate între formarea conceptelor și formularea, generalizarea și rezolvarea problemelor și a situațiilor-problemă (care presupun creativitate). În felul acesta se pune accent pe experiența directă a elevului, pe interesele sale spontane. Prioritar, elevul trebuie „învățat cum să învețe”.

Tipuri de lecții recomandate în care este valorificat principiul învățării active și conștiente:

1. Lecția de transmitere / însușire de (noi) cunoștințe;
2. Lecția de formare de priceperi și deprinderi;
3. Lecția mixtă (combinată).

DE REȚINUT ACESTE REPREZENTĂRI

Remarcă



Această imagine atenționează elevii despre unele informații suplimentare pe care elevii ar trebui să le cunoască sau la care ar trebui să atragă atenția.

Important



Această imagine atenționează elevii despre prezența unor informații importante, cu scopul de a dirija activitatea practică a elevilor.

Rezolvă individual



Această imagine atenționează elevii despre prezența unor sarcini ce trebuie realizate în mod individual, cu sau fără suportul profesorului.

Fișierele antet împreună cu funcțiile acestora sunt prezentate în anexe, precum și un model de testare finală la curs, asistat de calculator.

MEDII DE PROGRAMARE PENTRU LIMBAJUL C/C++

Introducere

Pentru a crea o aplicație de la zero, un dezvoltator ar trebui să se lupte mult. Deoarece pentru a crea o aplicație software ar fi nevoie de mulți pași începând de la Codificare, Compilare, Testare, Depanare, Construire și Implementare. Toate aceste faze diferite ar avea nevoie de pachete diferite pentru a fi instalate și dificil de întreținut de către un dezvoltator.

Deci, pentru a crește productivitatea și a accelera procesul de dezvoltare a software-ului, toate aceste pachete esențiale pentru dezvoltare sunt integrate sau combinate într-un mediu unic cunoscut sub numele de IDE (mediu de dezvoltare integrată), care a făcut dezvoltarea software-ului modern mai ușoară și mai rapidă.

Un IDE tipic trebuie să conțină caracteristici, cum ar fi evidențierea sintaxei, completarea codului automat, codul de salvare automată, căutarea codului, compilatorul, refactorizarea, depanarea, controlul versiunilor, construirea și implementarea etc.

Dacă doriți să fiți un programator C++, mai jos sunt câteva IDE gratuite pentru dezvoltarea de software:

1. **Visual Studio Code** este un editor de coduri open-source gratuit dezvoltat de Microsoft, care acceptă platforme precum Windows, Linux, Mac. Microsoft oferă o extensie pentru C++ numită „Microsoft C/C++”, care poate fi găsită cu ușurință prin căutarea în secțiunea de extensii. Există o altă extensie numită „cod runner” care permite să ruleze fragmente de cod ca ieșire. Mai multe detalii aici: <https://code.visualstudio.com/>
2. **Code :: blocks** este un IDE open-source și gratuit pentru C și C++ care poate fi utilizat pe mai multe platforme, construit în C++ folosind un instrument widget GUI numit „wxWidget”. Acest IDE este utilizat special

pentru programarea în C++ cu performanțe și caracteristici uniforme. Mai multe detalii aici: <https://www.codeblocks.org/>

3. **Clion** este un IDE modern de JetBrains, care este conceput pentru a rula C și C++ pe diverse platforme, cum ar fi Windows, Linux și MacOS. Oferă asistență inteligentă pentru asistența codului, astfel încât accentul principal trebuie să fie pe lucruri importante. Mai multe detalii aici: <https://www.jetbrains.com/clion/>
4. **Eclipse** este unul dintre cele mai populare și puternice IDE pentru dezvoltatorii C++. Este, de asemenea, open-source și gratuit de utilizat, cu un suport comunitar excelent. Suportă mai multe platforme precum Windows, Linux și MacOS și mulți utilizatori consideră că este ușor de utilizat. Mai multe detalii aici: <https://www.eclipse.org/>
5. **CodeLite** este unul dintre cele mai bune IDE C++ de utilizat, care este gratuit, open-source scris în C++ și specializat pentru a rula pe diverse platforme, cum ar fi Windows, MacOS, Linux. De asemenea, oferă multe funcții precum orice IDE C++. Mai multe detalii aici: <https://codelite.org/>

1.1 Prezentarea mediului de programare CodeBlocks 20

Ce este Code::Blocks ?

Code::Blocks este un mediu de programare sau un mediu integrat de dezvoltare (IDE) gratuit ce conține aproximativ tot ce are nevoie un programator pentru a crea aplicații. Voi enumera în continuare unele dintre facilitățile necesare:

- ◆ **editor de text** - necesar pentru scrierea programelor sau a codurilor sursă – fișiere cu extensia .c (pentru limbajul C) și .cpp (pentru limbajul C++);
- ◆ **compilatoare** - necesare pentru traducerea codurilor sursă în instrucțiuni pe care procesorul să le înțeleagă și crearea unui program executabil sau a unor fișiere compilate. Există mai multe compilatoare pentru limbajele C/C++, cel mai utilizat fiind gcc (GNU C Compiler)
- ◆ **biblioteci** - fișiere antet (header) care conțin descrierea anumitor operații, cele mai uzuale fiind citirea și afișarea datelor descrise în biblioteca iostream.

Orice fel de funcționalitate suplimentară poate fi introdusă prin instalarea de plugin-uri (ex: compilatoarele)!

De unde poate fi descărcat Code::Blocks?

Ultima versiune a softului Code::Blocks poate fi descărcată și utilizată gratuit de pe site-ul: <http://www.codeblocks.org/> din secțiunea Downloads, adică va fi accesă la acest link: <http://www.codeblocks.org/downloads> . După aceasta vom selecta **Download the binary release** . Alegeți versiunea pentru sistemul de operare **codeblocks-20.03mingw-setup.exe**. După selectarea opțiunii, veți obține o fereastră în care va trebui să accesați butonul verde de download.

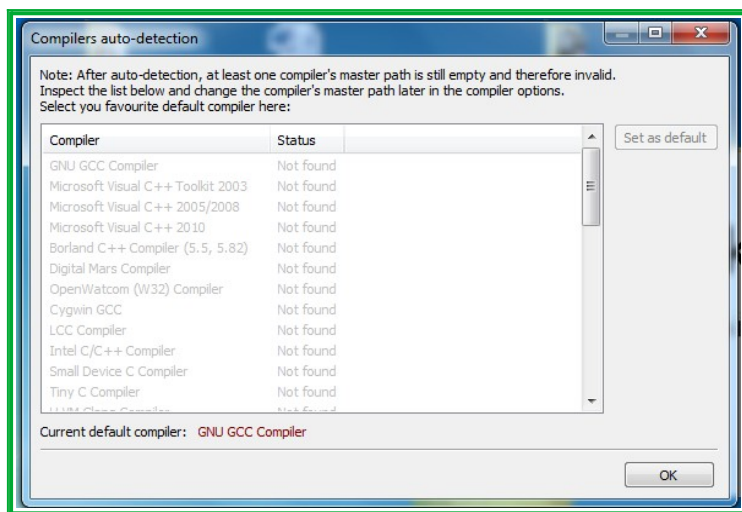


- ✓ Există, de asemenea, versiuni mai recente, de noapte, disponibile pe forumuri sau (pentru utilizatorii Ubuntu) în depozitul Ubuntu PPA.
- ✓ Un Changelog pentru 20.03, vă poate oferi o imagine de ansamblu asupra îmbunătățirilor și corecțiilor care au fost introduse în noua versiune.
- ✓ Versiunile implicite sunt pe 64 de biți (începând cu versiunea 20.03). De asemenea, este prezentă și versiunea de 32 de biți pentru confort.
- ✓ Fișierul codeblocks-20.03mingw-setup.exe include suplimentar compilatorul GCC/G++/GFortran și depanatorul GDB din proiectul MinGW-W64 (versiunea 8.1.0, 32/64 biți, SEH).
- ✓ Procesul de instalare pentru Code::Blocks poate fi uneori diferit, deoarece softul este mereu în actualizare.

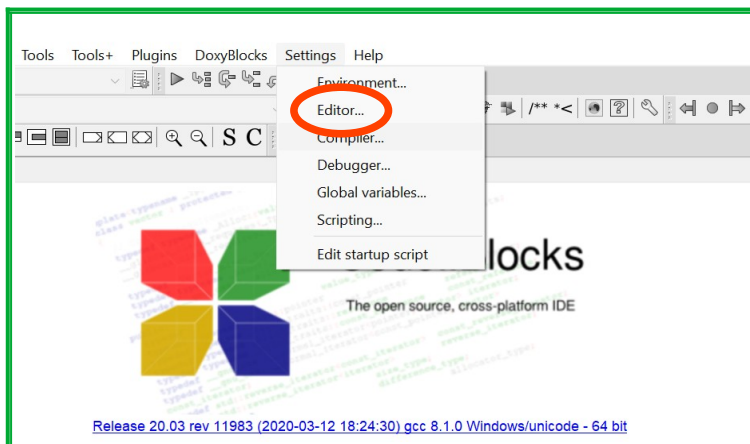
Cum poate fi instalat Code::Blocks?

1. Accesăm fișierul executabil, apoi accesăm **Next**, pentru a trece la următoarea etapă.
2. Este necesar să ne exprimăm acordul în legătură cu termenii și condițiile de utilizare ale IDE-ului Code::Blocks, accesăm **I Agree**.
3. Alegem componentele pe care dorim să le instalăm, apoi accesăm **Next**,
4. Alegem locația unde dorim să instalăm pachetele de program ale softului, această locație este recomandată a fi: **C:\Program Files\CodeBlocks**. După ce am ales locația, accesăm **Install** pentru a instala toate pachetele de program ale softului.
5. Observăm apariția unui shortcut pe Desktop.

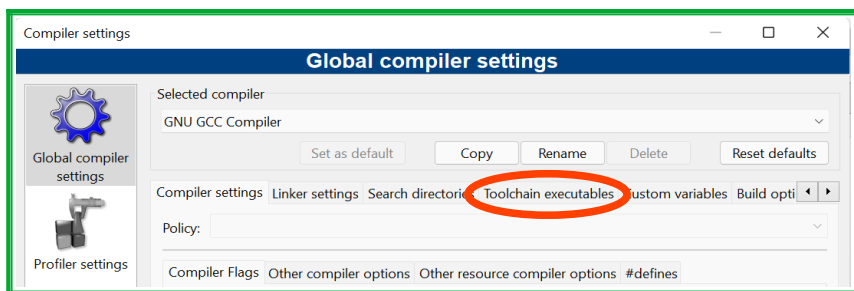
6. După instalare, este necesară selectarea compilatorului preferat. Implicit, acesta va fi GNU GCC Compiler.



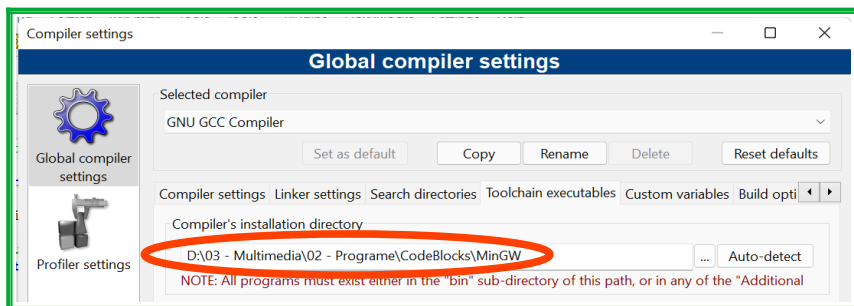
7. Selectarea compilatoarelor cu ajutorul cărora se vor executa programele sursă. Alegem opțiunea **Compiler** din meniul **Settings**.



8. La pasul respectiv, se deschide fereastra **Compiler settings** în care vom selecta opțiunea **Toolchain executables**.



9. Selectăm calea către compilatoare:

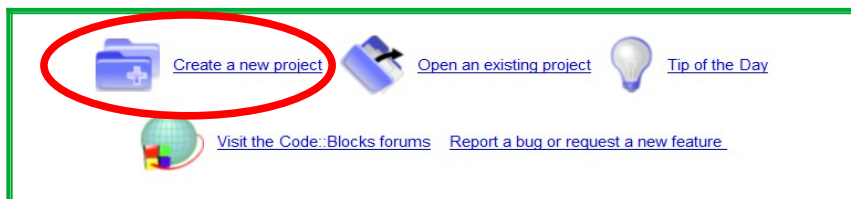


Cum poate fi creat primul program în Code::Blocks?

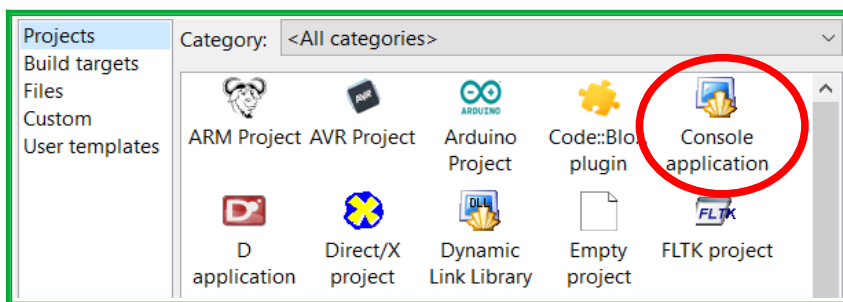
1. Efectuăm un dublu click pe iconița de pe desktop.



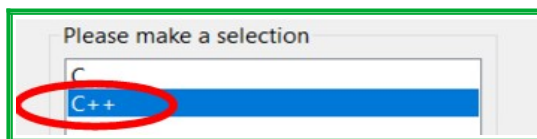
2. Din fereastra ce va apărea, vom selecta crearea proiect nou, **Create a new project**:



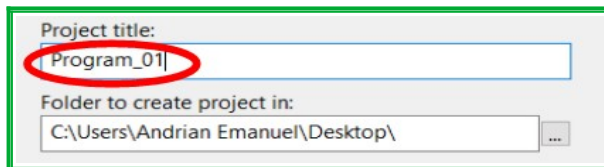
3. Alegem să creăm o aplicație de tip **Consolă**:



4. Selectăm limbajul (C++) în care vom scrie aplicația de consolă:



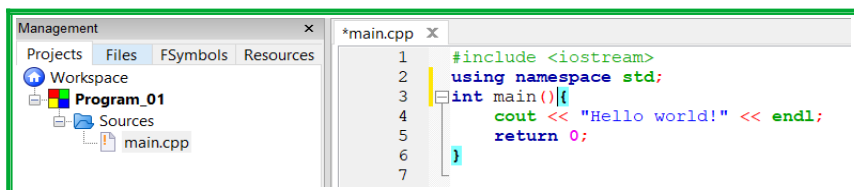
5. Stabilim o denumire proiectului nostru, **Program_01**:



6. Alegem compilatorul de care avem nevoie pentru proiectul nostru:

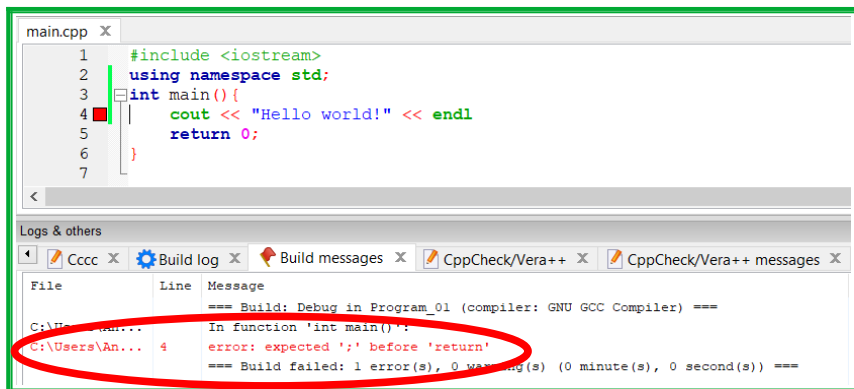


7. Editarea programului se va face în fișierul **main.cpp** , creat implicit și plasat în folderul **Sources**. Scrierea primului program C++:



8. Opțiunile pentru compilarea și execuția programului se găsesc în meniul **Build**. Vom alege opțiunea **Build and run (F9)**. Rezultatul va fi afișarea la ecran (în consolă) a mesajului scris cu ajutorul instrucțiunii de afișare **cout**.

Exemplu de secvență de cod C++ cu o eroare de sintaxă în rândul 4:



1.2 Prezentarea unui IDE pentru smartphone

Cxxdroid este un compiler IDE ce poate fi instalat ușor pe orice dispozitiv mobil cu sistem de operare Android, cum ar fi tabletă sau smartphone. Să analizăm unele caracteristici speciale ale sale:

- ◆ *Compiler C/C++ offline: nu este necesar Internet pentru a rula programe C/C++;*
- ◆ *Manager de pachete și un depozit personalizat cu pachete predefinite pentru biblioteci obișnuite, cum ar fi Boost, SQLite, ncurses, libcurl etc.; Sunt disponibile, de asemenea, biblioteci grafice precum SDL2, SFML* și Allegro*;*
- ◆ *Exemple disponibile imediat pentru o învățare mai rapidă;*
- ◆ *Emulator terminal cu suport complet; Modul de interpretare C/C++ (REPL) bazat pe CERN Cling este de asemenea disponibil;*
- ◆ *Performanță remarcabilă cu tehnologia avansată de stocare în cache a compilatorului: de până la 33 de ori mai rapidă atunci când este utilizată biblioteca Boost, viteză medie de 3 ori;*
- ◆ *Arhitectură curată și matură: codul este analizat și compilat cu același compilator, iar IDE-ul nu se blochează complet din cauza erorilor de runtime din programe;*
- ◆ *Interfața de utilizare concepută având în vedere viteza și gradul de utilizare;*
- ◆ *Adevărat compiler: nu sunt implicați interpreți pe bază de java (sau chiar javascript), chiar și limbajul de asamblare în linie este acceptat (sintaxă Clang).*

Să analizăm unele caracteristici de editare:

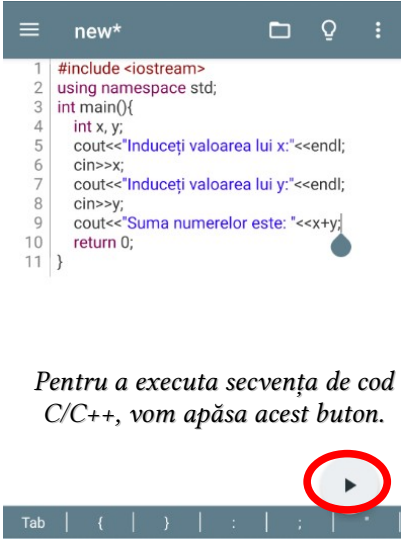
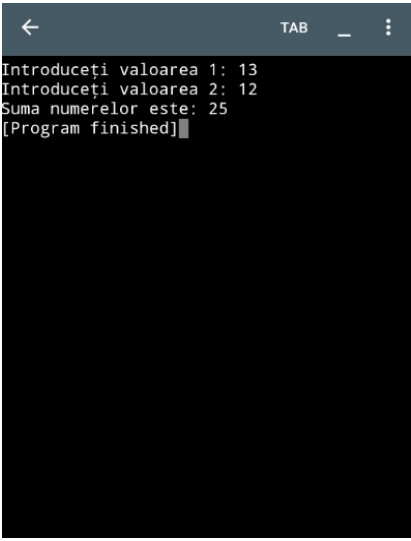
- *Predicție de cod în timp real, indentare automată și analiză de cod la fel ca în orice IDE real*.*
- *Bara de tastatură extinsă cu toate simbolurile de care aveți nevoie pentru a programa în C/C++;*
- *Evidențierea sintaxei și teme;*
- *Partajare cu un singur clic pe Pastebin.*





- ✓ Funcțiile marcate cu asterisc sunt disponibile numai în versiunea Premium.
- ✓ Cxxdroid necesită cel puțin 150 MB memorie internă. Se recomandă 200 MB+. Mai mult dacă utilizați biblioteci mari precum Boost.
- ✓ Pentru a executa secvența de cod scrisă în limbajul C/C++, este necesar doar să apăsați butonul de execuție evidențiat în imaginea de mai jos.

Un exemplu foarte simplu de secvență de cod scrisă în limbajul C++ cu ajutorul softului prezentat mai sus arată astfel:

Fereastra de scriere a codului	Fereastra de execuție a codului
 <pre>1 #include <iostream> 2 using namespace std; 3 int main(){ 4 int x, y; 5 cout<<"Introduceți valoarea lui x:"<<endl; 6 cin>>x; 7 cout<<"Introduceți valoarea lui y:"<<endl; 8 cin>>y; 9 cout<<"Suma numerelor este: "<<x+y; 10 return 0; 11 }</pre> Pentru a executa secvența de cod C/C++, vom apăsa acest buton.	 <pre>Introduceți valoarea 1: 13 Introduceți valoarea 2: 12 Suma numerelor este: 25 [Program finished]</pre>

Un alt exemplu de secvență de cod scrisă în limbajul C++ cu ajutorul softului prezentat mai sus arată astfel:

Fereastra de scriere a codului	Fereastra de execuție a codului
A play button icon is visible at the bottom right of the editor window."/>	

1.3 Prezentarea unui IDE pentru Linux

1. Geany este un editor de text puternic, care oferă o mulțime de funcții utile, fără a vă împiedica fluxul de lucru. Este tradus în peste 40 de limbi și are suport încorporat pentru mai mult de 50 de limbaje de programare.

Geany este un mediu de dezvoltare integrat mic și ușor. A fost dezvoltat pentru a oferi un IDE mic și rapid, care are doar câteva dependențe de alte pachete. Un alt obiectiv a fost să fie cât mai independent posibil de un mediu special de birou, cum ar fi KDE sau GNOME, Geany necesită doar bibliotecile de execuție GTK2. Se cunoaște că Geany rulează sub Linux, FreeBSD, Net-BSD, OpenBSD, MacOS X, AIX v5.3, Solaris Express și Windows. Mai general, ar trebui să ruleze pe fiecare platformă, care acceptă bibliotecile GTK. Numai portului Windows din Geany îi lipsesc unele caracteristici. Pentru mai multe detalii, accesați: <https://www.geany.org/>

Caracteristici importante:

1. Evidențierea sintaxei;
2. Plierea codului;
3. Completarea automată a numelui simbolului;
4. Construiți finalizare / fragmente;
5. Închiderea automată a etichetelor XML și HTML;
6. Sfaturi pentru apel;
7. Multe tipuri de fișiere acceptate, inclusiv C, java, PHP, HTML, Perl, Python, Pascal (listă completă);
8. Liste de simboluri;
9. Navigare cod;
10. Construiți un sistem pentru a compila și executa codul;
11. Management de proiect simplu;
12. Interfața pluginului (consultați pluginurile).

Instalarea pachetului Geany pe Ubuntu pas cu pas

Instrucțiuni de instalare rapidă a geany pe serverul Ubuntu. Este foarte ușor! Pur și simplu scrieți comanda în terminalul dvs. de linie de comandă folosind managerul de pachete APT încorporat. Vedeți mai jos instrucțiunile rapide pas cu pas ale comenzilor SSH, Copiere / Lipire pentru a evita ortografia greșită sau instalarea accidentală a unui alt pachet.

Pasul 1: Rulați comanda de actualizare pentru a actualiza depozitele de pachete și pentru a obține cele mai recente informații despre pachet.

sudo apt-get update -y

Pasul 2: Rulați comanda de instalare cu flagul de stare -y, pentru a instala rapid pachetele și dependențele.

sudo apt-get install -y geany

Pasul 3: Verificați jurnalele de sistem (system logs) pentru a confirma lipsa erorilor asociate. Puteți utiliza **ZoomAdmin** pentru a verifica jurnalele, serverele manager, găzdui mai multe site-uri web și aplicații pe serverele dvs. și multe altele. Aplicațiile rulează în containere docker.

2. Brackets este un editor de text modern și open-source conceput special pentru proiectarea și dezvoltarea web. Este foarte extensibil prin intermediul pluginurilor, de aceea programatorii C/C++ îl pot folosi instalând extensia pachetului C/C++/Objective-C. Acest pachet este conceput pentru a îmbunătăți scrierea codului C/C++ și pentru a oferi caracteristici asemănătoare IDE-urilor. Pentru mai multe detalii, accesați: <http://brackets.io/>

Instalați Brackets utilizând Snap pas cu pas

Începând cu Ubuntu 16.04, Brackets este disponibil ca aplicație Snap. Îl puteți găsi în Software Center și îl puteți instala de acolo în câteva clicuri.

Dacă doriți să mergeți pe linia de comandă, puteți utiliza comenzile Snap pentru a instala Brackets. Acest lucru va funcționa pentru toate distribuțiile Linux care acceptă ambalarea (pachetele) Snap.

Pasul 1: Asigurați-vă că ați instalat daemon `snapd` pe sistemul dvs. Folosiți comanda de mai jos pentru a instala. Nu vă faceți griji dacă este deja instalat. Doar executați comanda:

```
sudo apt-get install snapd
```

Pasul 2: După ce v-ați asigurat că `snapd` este instalat pe sistemul dvs. Linux, instalați Brackets utilizând comanda:

```
sudo snap install brackets
```

APLICAȚII FUNDAMENTALE

2.1 Aplicarea transformărilor elementare



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit funcțiile de intrare/ieșire și tipurile de operatori.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 2.1 - 01

Elaborați un cod C/C++ care va afișa un triunghi de numere simetric, centrat și ascendent.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> #include<math.h> void main(){ printf(" TRIUNGHI \n"); printf(" 1 \n"); printf(" 121 \n"); printf(" 12321 \n"); printf(" 1234321 \n"); printf(" 123454321 \n"); }</pre>	<pre>#include<iostream> using namespace std; int main(){ cout<<" TRIUNGHI "<<endl; cout<<" 1 "<<endl; cout<<" 121 "<<endl; cout<<" 12321 "<<endl; cout<<" 1234321 "<<endl; cout<<" 123454321 "<<endl; }</pre>

Problema 2.1 - 02

Elaborați un cod C/C++ care va afișa un ghiveci cu flori. Pentru a realiza un asemenea desen vom aplica simbolurile textuale.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    cout<<"Proiectul FLOARE\n\n";
    cout<<"      (^)      "<<endl;
    cout<<"    (^)  ( o )(^)  "<<endl;
    cout<<"  ( o )  ( _ )( o ) "<<endl;
    cout<<"    ( _ )  ( ) ( _ ) "<<endl;
    cout<<"      ( ) ( o )( )  "<<endl;
    cout<<"    ( o ) ( _ )( o ) "<<endl;
    cout<<"    ( _ )\ \ | / ( _ ) "<<endl;
    cout<<"    _ \ _ \ \ | / _ / _ "<<endl;
    cout<<"      )  _____ ( "<<endl;
    cout<<"    ( ---- ( ) ---- ) "<<endl;
    cout<<"      \ \ -- ( _ @ ) -- / "<<endl;
    cout<<"      \ \   ( _ )   / "<<endl;
    cout<<"      '-----'      "<<endl;
    cout<<endl;
}
```

Problema 2.1 - 03

Elaborați un cod C/C++ care va afișa un catren dintr-o poezie. Specificați titlul poeziei în partea de sus, iar autorul poeziei în partea de jos.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    cout<<"Poezia \"Somnoroase pasarele\"!"<<endl;
    cout<<"\n\tSomnoroase pasarele";
    cout<<"\n\tPe la cuiburi se aduna";
    cout<<"\n\tSe ascund in ramurile";
    cout<<"\n\tNoapte buna! Noapte buna!"<<endl;
    cout<<"\nautor M. Eminescu" << endl;
}
```

Problema 2.1 - 04

Elaborați un cod C/C++ care va implementa operatorii aritmetici în C++: „+”, „-”, „*”, „/” și „%”, cu ajutorul a două numere întregi A și B introduse de la tastatură.

Implementarea C	Implementarea C++
<pre>#include <stdio.h> #include <stdlib.h> int main(){ int a, b; //declarăm a și b printf("Introducem A: "); scanf("%d",&a); //citim a printf("Introducem B: "); scanf("%d",&b); //citim b printf("\nSuma:\t\t"); printf("%d",a+b); printf("\nDiferenta: \t"); printf("%d",a-b); printf("\nProdusul:\t"); printf("%d",a*b); printf("\nCatul:\t\t"); printf("%d",a/b); printf("\nRestul:\t\t"); printf("%d",a%b); return 0; }</pre>	<pre>#include <iostream> using namespace std; int main(){ int a, b; //declarăm a și b cout<<"Introducem A: "; cin>>a; // citim a cout<<"Introducem B: "; cin>>b; // citim b cout<<"\nSuma este:\t\t"; cout<<a+b; cout<<"\nDiferenta este: \t"; cout<<a-b; cout<<"\nProdusul este:\t\t"; cout<<a*b; cout<<"\nCatul este:\t\t"; cout<<a/b; cout<<"\nRestul este:\t\t"; cout<<a%b; return 0; }</pre>

Problema 2.1 - 05

Elaborați un cod C/C++ care va implementa operatorii de atribuire în C++: „=”, „+=”, „-=”, „*=”, „/=” și „%=", cu ajutorul a două numere întregi A și B introduse de la tastatură.

Implementarea C	Implementarea C++
<pre>#include <stdlib.h> int main(){ int a, b; //declarăm a și b printf("Introducem A & B: "); scanf("%d %d",&a,&b); // citim b = a; printf("= Solutia: %i\n",b); b += a; printf("+= Solutia: %i\n",b); b -= a; printf("-= Solutia: %i\n",b); b *= a; printf("*= Solutia: %i\n",b); }</pre>	<pre>#include <iostream> using namespace std; int main(){ int a, b; //declarăm a și b cout<<"Introducem A & B: "; cin>>a>>b; // citim cout<<endl; b = a; cout<<"= Rezultat: "<<b<<endl; b += a; cout<<"+= Rezultat: "<<b<<endl; b -= a; cout<<"-= Rezultat: "<<b<<endl; }</pre>

<pre> b /= a; printf("/= Solutia: %i\n",b); b %= a; printf("%%= Solutia: %i\n",b); return 0; } </pre>	<pre> b *= a; cout<<"*= Rezultat: "<<b<<endl; b /= a; cout<<"/= Rezultat: "<<b<<endl; b %= a; cout<<"%= Rezultat: "<<b<<endl; return 0; } </pre>
---	--

Problema 2.1 - 06

Elaborați un cod C/C++ care va implementa operatorii de auto incrementare / decrementare, cu ajutorul a două numere întregi A și B introduse de la tastatură.

Implementarea C	Implementarea C++
<pre> #include <stdlib.h> #include <stdio.h> int main(){ int a, b; //declaram a și b printf("Introducem A & B: "); scanf("%d %d",&a,&b); // citim printf("\n"); printf("%d ++ \tRezultat: ",a); a++; // incrementare printf("%d \n",a); printf("%d -- \tRezultat: ",b); b--; // decrementare printf("%d \n",b); return 0; } </pre>	<pre> #include <iostream> using namespace std; int main(){ int a, b; //declaram a & b cout<<"Introduceti A & B: "; cin>>a>>b; // citim a & b cout<<endl; cout<<a<<"++ \tRezultat: "; a++; // incrementare cout<<a<<endl; cout<<b<<"-- \tRezultat: "; b--; // decrementare cout<<b<<endl; return 0; } </pre>

Problema 2.1 - 07

Elaborați un cod C/C++ care va implementa operatorii logici în C++: „&&”, „||” și „!”, cu ajutorul a două numere booleene B1 și B2 introduse de la tastatură.

Implementarea C	Implementarea C++
<pre> #include <stdio.h> #include <stdbool.h> int main(){ </pre>	<pre> #include <iostream> using namespace std; int main(){ </pre>

<pre> bool b1, b2; //declarăm b1 & b2 printf("Introducem B1 & B2: "); scanf("%d %d",&b1,&b2); // b1= 0 (true), b2= 1 (false) printf("b1 && b2:\t%d",b1&& b2); printf("\n"); printf("b1 b2:\t%d",b1 b2); printf("\n"); printf("!(b1 && b2):\t"); printf("%d",!(b1&&b2)); return 0; } </pre>	<pre> bool b1, b2; //declarăm b1 & b2 cout<<"Introduceti B1 & B2: "; cin>>b1>>b2; // citim b1 & b2 // b1= 0 (true), b2= 1 (false) cout<<endl; cout<<"\nb1 && b2: \t"; cout<<(b1&&b2); cout<<"\nb1 b2: \t"; cout<<(b1 b2); cout<<"\n!(b1 && b2): \t"; cout<<!(b1&&b2); return 0; } </pre>
---	---

Problema 2.1 - 08

Elaborați un cod C/C++ care va implementa operatorii de relaționare în C++: „=”, „!=”, „>”, „<”, „>=”, „<=” și „<”, cu ajutorul a două numere întregi A și B introduse de la tastatură.

Implementarea C	Implementarea C++
<pre> #include <stdio.h> #include <stdbool.h> int main(){ int a,b; //declarăm a & b printf("Introducem A & B: "); scanf("%d %d",&a,&b); // 0 = true, 1 = false // Implementarea operatorilor printf("\na < b:\t%d", (a<b)); printf("\na > b:\t%d", (a>b)); printf("\na <= b:\t%d", (a<=b)); printf("\na >= b:\t%d", (a>=b)); printf("\na != b:\t%d", (a!=b)); printf("\na == b:\t%d", (a==b)); return 0; } </pre>	<pre> #include <iostream> using namespace std; int main(){ int a, b; //declarăm a & b cout<<"Introduceti A & B: "; cin>>a>>b; // citim a & b cout<<endl; cout<<a<<" < "<<b<<"\t=>\t"; cout<<(a<b)<<endl; cout<<a<<" > "<<b<<"\t=>\t"; cout<<(a>b)<<endl; cout<<a<<" <= "<<b<<"\t=>\t"; cout<<(a<=b)<<endl; cout<<a<<" >= "<<b<<"\t=>\t"; cout<<(a>=b)<<endl; cout<<a<<" != "<<b<<"\t=>\t"; cout<<(a!=b)<<endl; cout<<a<<" == "<<b<<"\t=>\t"; cout<<(a==b)<<endl; return 0; } </pre>

Problema 2.1 - 09

Elaborați un cod C/C++ care va implementa operatorii bit-bit în C++: „&”, „|”, „^”, „~”, „<<” și „>>”, cu ajutorul a două numere întregi A și B introduse de la tastatură.

Implementarea C	Implementarea C++
<pre>#include <stdio.h> int main(){ int a,b; // 11 = 00001011 // 22 = 00010110 printf("Introducem A & B: "); scanf("%d %d",&a,&b); // Implementarea operatorilor printf("\na & b:\t%d", (a&b)); printf("\na b:\t%d", (a b)); printf("\na ^ b:\t%d", (a^b)); printf("\n~ a:\t%d", (~a)); printf("\na << 3:\t%d", (a<<3)); printf("\na >> 2:\t%d", (a>>2)); return 0; }</pre>	<pre>#include <iostream> using namespace std; int main(){ int a,b,rez=0; // 11 = 00001011 // 22 = 00010110 cout<<"Introduceti A & B: "; cin>>a>>b; // citim a & b rez = a & b; cout<<"\na & b: \t\t"<<rez; rez = a b; cout<<"\na b: \t\t"<<rez; rez = a ^ b; cout<<"\na ^ b: \t\t"<<rez; rez = ~a; cout<<"\n~a: \t\t"<<rez; rez = a << 3; cout<<"\na << 3: \t"<<rez; rez = b >> 2; cout<<"\na >> 2: \t"<<rez; return 0; }</pre>

2.2 Aplicarea formulelor matematice



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica unele transformări elementare ale formulelor matematice din diverse domenii: matematică, fizică, chimie, etc.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 2.2 - 01

Elaborați un cod C/C++ care va afișa la ecran rezultatul unei expresii matematice. În acest exemplu litera A, reprezintă numărul de ordine al elevului în registrul grupei. Introduceți la început valoarea lui A, apoi afișați soluția.

$$f(x) = \sin\left(\frac{5\pi}{A}\right) + \cos\left(\frac{3\pi}{A}\right)$$

Implementarea C	Implementarea C++
<pre>#include <stdio.h> #include <stdlib.h> #include <math.h> #define Pi 3.14 // constanta int main(){ int a; // declarăm A float rez; printf("Introduceti A: "); scanf("%d",&a); // citim A rez=sin(5*Pi/a)+sin(3*Pi/a); printf("Solutia expresiei: "); printf("%f",rez); }</pre>	<pre>#include <iostream> #include <cmath> #define Pi 3.14 // constanta using namespace std; int main(){ int a; // declarăm A float rez; cout<<"Introduceti A: "; cin>>a; // citim A rez=sin(5*Pi/a)+cos(3*Pi/a); cout<<"Solutia expresiei: "; cout<<rez; }</pre>

Problema 2.2 - 02

Elaborați un cod C/C++ care va afișa la ecran rezultatul unei expresii matematice. În acest exemplu litera A, reprezintă numărul de ordine al elevului în registrul grupei. Introduceți la început valoarea lui A, apoi afișați soluția.

$$\frac{9^2 + \sqrt{(8+A)^3}}{A \cdot (2+A)} - \frac{3^2 + \sqrt{(7+A)^3}}{A \cdot (5+A)}$$

Pentru a rezolva corect această problemă, vom aplica unele transformări elementare. Pentru început, vom determina numitorul comun, apoi vom efectua calculele corespunzătoare pentru a obține doar o singură fracție. Numitorul și numărătorul îl vom nota printr-o variabilă n1, respectiv n2:

- $m=2+a$; $n=5+a$; $\Rightarrow n1=a*m*n$;
- $x=((9*9)+\text{sqrt}(\text{pow}(8+a,3)))^*n$; $y=((3*3)+\text{sqrt}(\text{pow}(7+a,3)))^*m$; $\Rightarrow n2=x-y$;

Implementarea C	Implementarea C++
<pre>#include <stdio.h> #include <stdlib.h> #include <math.h> int main(){ int a; // declarăm A // numitor și numărător float n1, n2; float x,y; printf("Introduceti A: "); scanf("%d",&a); // citim A int m=2+a, n=5+a; // notatii n1=a*m*n; // 9*9=pow(9,2); 3*3=pow(3,2); x=((9*9)+sqrt(pow(8+a,3)))^*n; y=((3*3)+sqrt(pow(7+a,3)))^*m; n2=x-y; printf("Solutia este: "); printf("%f",n2/n1); }</pre>	<pre>#include <iostream> #include <cmath> using namespace std; int main(){ int a; // declarăm A // numitor și numărător float n1, n2; float x,y; cout<<"Introduceti A: "; cin>>a; // citim A int m=2+a, n=5+a; // notatii n1=a*m*n; // 9*9=pow(9,2); 3*3=pow(3,2); x=((9*9)+sqrt(pow(8+a,3)))^*n; y=((3*3)+sqrt(pow(7+a,3)))^*m; n2=x-y; cout<<"Solutia este: "; cout<<n2/n1; }</pre>

Problema 2.2 - 03

Elaborați un cod C/C++ care va afișa numărul de laturi ale poligonului, dacă se cunoaște că suma unghiurilor unui poligon regulat este S^0 .

Implementarea C	Implementarea C++
<pre>#include<stdio.h> void main(){</pre>	<pre>#include <iostream> using namespace std;</pre>

<pre> int S; // Introducem suma printf("Care este S? "); scanf("%d",&S); // citim S // Formula de calcul: // S=180*(n-2); printf("\nNumarul de laturi:"); printf("%d",(S/180)+2); } </pre>	<pre> int main(){ int S; // Introducem suma cout<<"Introduceti S:"; cin>>S; // citim S // Formula de calcul: // S=180*(n-2); cout<<"\nNumarul de laturi:"; cout<<(S/180)+2; } </pre>
--	--

Problema 2.2 - 04

Elaborați un cod C/C++ care va afișa timpul T exprimat în minute, ore și zile, dacă se cunoaște că $T \in [0, 1000000]$.

Implementarea C	Implementarea C++
<pre> #include<stdio.h> #include<math.h> void main(){ int s; // declarăm s = secunde float m,o,z;//minute,ore,zile printf("Introduceti s: "); scanf("%d",&s); // citim sec. // calculam min., sec., zile m=s/60; o=m/60; z=o/24; printf("\nMinute: %f",m); printf("\nOre: %t%f",o); printf("\nZile: %t%f",z); } </pre>	<pre> #include <iostream> using namespace std; int main(){ int s; // declarăm s = secunde float m,o,z;//minute,ore,zile cout<<"Introduceti s: "; cin>>s; // citim sec. // calculam min., sec., zile m=s/60; o=m/60; z=o/24; cout<<"\nMinute: "<<m; cout<<"\nOre: %t"<<o; cout<<"\nZile: %t"<<z; } </pre>

Problema 2.2 - 05

Elaborați un cod C/C++ care va afișa latura și perimetrul figurilor geometrice, dacă se cunoaște R – raza cercului circumscris triunghiului echilateral și a pătratului.

Implementarea C	Implementarea C++
<pre> #include<stdio.h> #include<math.h> void main(){ float R; // declarăm R </pre>	<pre> #include <iostream> #include <cmath> using namespace std; int main(){ </pre>

<pre> printf("Introduceti R: "); scanf("%f",&R); // citim R // Triunghiul echilateral printf("\n"); printf("Latura triunghi: "); printf("\t%f",R*sqrt(3)); printf("\nPerimetru triunghi:"); printf("\t%f",4*R*sqrt(3)); // Patratul printf("\n"); printf("\nLatura patrat: "); printf("\t\t%f", (2*R)/sqrt(2)); printf("\nPerimetru patrat: "); printf("\t%f", 4*(2*R)/sqrt(2)); } </pre>	<pre> float R; // declarăm R cout<<"Introduceti R: "; cin>>R; // citim R // Triunghiul echilateral cout<<endl; cout<<"Latura triunghi: \t"; cout<<R*sqrt(3); cout<<"\nPerimetru triunghi:\t"; cout<<3*R*sqrt(3); // Patratul cout<<endl; cout<<"\nLatura patrat: \t\t"; cout<<(2*R)/sqrt(2); cout<<"\nPerimetru patrat:\t"; cout<<4*(2*R)/sqrt(2); } </pre>
---	---

Problema 2.2 - 06

Elaborați un cod C/C++ care va afișa raportul dintre volumul conului și al cilindrului, dacă aceste figuri sunt circular drepte, R – raza bazei conului și al cilindrului, iar H – înălțimea acestor figuri. Valorile pentru R și H se vor citi de la tastatură.

Implementarea C	Implementarea C++
<pre> #include<stdio.h> #include<math.h> #define Pi 3.14 // constanta void main(){ float R; // declarăm R int H; // declarăm H printf("Introduceti R: "); scanf("%f",&R); // citim R printf("Introduceti H: "); scanf("%d",&H); // citim H // Volumul cilindrului printf("Volum cilindru: "); float Vc=Pi*pow(R,2)*H; printf("%f",Vc); // Volumul conului printf("\nVolum con:"); printf("\t%f",Vc/3); // Raportul V_con/V_cil </pre>	<pre> #include <iostream> #include <cmath> #define Pi 3.14 // constanta using namespace std; int main(){ float R; // declarăm R int H; // declarăm H cout<<"Introduceti R: "; cin>>R; // citim R cout<<"Introduceti H: "; cin>>H; // citim H // Volumul cilindrului cout<<"\nVolum cilindru: "; float Vc=Pi*pow(R,2)*H; cout<<Vc; // Volumul conului cout<<"\nVolum con: \t"; cout<<Vc/3; </pre>

<pre>printf("\nRaportul: "); printf("\t%f", (Vc/3)/(Vc)); }</pre>	<pre>// Raportul V_con/V_cil cout<<"\nRaportul: \t"; cout<<((Vc/3)/(Vc)); }</pre>
---	---

Problema 2.2 - 07

Elaborați un cod C/C++ care va afișa valoarea forței (F) care, deplasând rectiliniu un corp pe distanța de D cm, a efectuat un lucru mecanic egal cu L Jouli.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> void main(){ int L; // declarăm L float D; // declarăm D // Introducem distanța în cm printf("Care este distanța?"); scanf("%f",&D); // citim D // Introducem lucrul în Jouli printf("Care este lucrul?"); scanf("%d",&L); // citim L // Formula de calcul: L=F*D printf("\nForta este: "); printf("%f N",L/(D/100)); }</pre>	<pre>#include <iostream> using namespace std; int main(){ int L; // declarăm L float D; // declarăm D // Introducem distanța în cm cout<<"Introduceti distanța:"; cin>>D; // citim D // Introducem lucrul în Jouli cout<<"Introduceti lucrul: "; cin>>L; // citim L // Formula de calcul: L=F*D cout<<"\nForta este: "; cout<<L/(D/100)<<"N."; }</pre>

Problema 2.2 - 08

Elaborați un cod C/C++ care va afișa valoarea presiunii necesare pentru reducerea volumului la V_2 dm³ fără variația temperaturii, dacă se cunoaște că o masă de gaz ocupă un volum de V_1 dm³ la presiunea de P_1 Pa.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> void main(){ int V1, V2, P1; printf("Care este V1? "); scanf("%d",&V1); // citim V1 printf("Care este V2? "); scanf("%d",&V2); // citim V2</pre>	<pre>#include <iostream> using namespace std; int main(){ int V1, V2, P1; cout<<"Introduceti V1:"; cin>>V1; // citim V1 cout<<"Introduceti V2:";</pre>

<pre>// Introducem P1 printf("Care este P1? "); scanf("%d",&P1); // citim P1 // Formula de calcul: // P1*V1=P2*V2 printf("\nPresiune va fi:"); printf("%d Pa",(P1*V1)/V2); }</pre>	<pre>cin>>V2; // citim V2 // Introducem P1 cout<<"Introduceti P1:"; cin>>P1; // citim P1 // Formula de calcul: // P1*V1=P2*V2 cout<<"\nPresiune va fi:"; cout<<(P1*V1)/V2<<"Pa."; }</pre>
--	---

Problema 2.2 - 09

Elaborați un cod C/C++ care va afișa valoarea ariei unui disc (A_d) în care este înscris un triunghi echilateral cu aria de A_t cm².

Pentru a rezolva corect această problemă, vom aplica unele transformări elementare. Pentru început vom prezenta unele formule matematice pe care le vom implementa în această rezolvare:

$$A_d = \pi R^2; R = \frac{2 \cdot h}{3}; A_t = \frac{l^2 \cdot \sqrt{3}}{4} \Rightarrow l^2 = \frac{A_t \cdot 4}{\sqrt{3}} \quad (1)$$

$$l = \sqrt{\frac{A_t \cdot 4}{\sqrt{3}}}; h = \frac{l \cdot \sqrt{3}}{2} \Rightarrow R = \frac{2 \cdot h}{3} \quad (2)$$

Conform seturilor de formule (1) și (2), vom obține soluția problemei.

Implementarea C	Implementarea C++
<pre>#include<math.h> #define Pi 3.14 // constanta void main(){ float At,l,h,R; // variabilele printf("Care este V1? "); scanf("%f",&At); // citim At l=sqrt((At*4)/sqrt(3)); h=(l*sqrt(3))/2; R=(2*h)/3; printf("\nAria discului:"); printf("%f cm.p",Pi*pow(R,2)); }</pre>	<pre>#include <iostream> #include <cmath> #define Pi 3.14 // constanta using namespace std; int main(){ float At,l,h,R; // variabilele cout<<"Introduceti At:"; cin>>At; // citim At l=sqrt((At*4)/sqrt(3)); h=(l*sqrt(3))/2; R=(2*h)/3; cout<<"\nAria discului:"; cout<<Pi*pow(R,2)<<" cm.p"; }</pre>

2.3 Aplicarea operatorului condițional



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica operatorul condițional împreună cu unii operatori logici.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 2.3 - 01

Folosind operatorul condițional, elaborați un cod C/C++ care va determina dacă numărul unui bilet C poate fi câștigător. Un număr de bilet C poate fi câștigător, dacă este mai mic decât jumătatea unui număr bingo B. Toate variabilele se vor citi de la tastatură, iar rezultatul se va afișa sub forma „Da” sau „Nu”.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> int main(){ int b,c; // Introducem datele printf("Introducem B si C:\n"); scanf("%d %d",&b,&c); // Aplicarea operatorului printf(c<(b/2)?"Da":"Nu"); }</pre>	<pre>#include <iostream> using namespace std; int main(){ int b,c; // Introducem datele cout<<"Introducem B si C:\n"; cin>>b>>c; // Aplicarea operatorului cout<<(c<(b/2)?"Da":"Nu"); }</pre>

Problema 2.3 - 02

Folosind operatorul condițional, elaborați un cod C/C++ care va determina dacă o valoare X aparține intervalului $[a, b]$. Toate variabilele se vor citi de la tastatură, iar rezultatul se va afișa sub forma „Da” sau „Nu”.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> #include<stdbool.h></pre>	<pre>#include <iostream> using namespace std;</pre>

<pre> int main(){ int x,a,b; bool rez; // Introducem datele printf("Introducem datele:\n"); scanf("%d %d %d",&x,&a,&b); // Aplicarea operatorului rez=(x>=a && x<=b); printf(rez?"- Da":"- Nu"); } </pre>	<pre> int main(){ int x,a,b; bool rez; // Introducem datele cout<<"Introducem datele:\n"; cin>>x>>a>>b; // Aplicarea operatorului rez=(x>=a && x<=b); cout<<(rez?"- Da":"- Nu"); } </pre>
---	---

Problema 2.3 - 03

Folosind operatorul condițional, elaborați un cod C/C++ care va determina dacă trei numere reale a , b și c pot fi lungimile laturilor unui triunghi.

Implementarea C	Implementarea C++
<pre> #include<stdio.h> #include<stdbool.h> int main(){ float a,b,c; bool rez; // Introducem laturile printf("Care sunt laturile?"); scanf("%f %f %f",&a,&b,&c); // Aplicarea operatorului rez=(a+b>c && a+c>b && b+c>a); printf(rez?"Da":"Nu"); } </pre>	<pre> #include <iostream> using namespace std; int main(){ float a,b,c; // Introducem laturile cout<<"Introducem datele:"; cin>>a>>b>>c; // Aplicarea operatorului cout<<(a+b>c && a+c>b && b+c>a?"Da":"Nu"); } </pre>

Problema 2.3 - 04

Folosind operatorul condițional, elaborați un cod C/C++ care va determina dacă un an este bisect. Rezultatul se va afișa sub forma „Da” sau „Nu”.

Implementarea C	Implementarea C++
<pre> #include<stdio.h> #include<stdbool.h> int main(){ int an; bool rez1,rez2; // Introducem anul printf("Introducem anul:\n"); scanf("%d",&an); } </pre>	<pre> #include <iostream> using namespace std; int main(){ int an; bool rez1,rez2; // Introducem anul cout<<"Introducem anul:\n"; cin>>an; } </pre>

<pre>// Aplicarea operatorului rez1=(an%4==0 && an%100!=0); rez2=(rez1 an%400==0); printf(rez2?"- Da:"- Nu"); }</pre>	<pre>// Aplicarea operatorului rez1=(an%4==0 && an%100!=0); rez2=(rez1 an%400==0); cout<<(rez2?"- Da:"- Nu); }</pre>
--	---

Problema 2.3 - 05

Folosind operatorul condițional, elaborați un cod C/C++ care va determina de câte sticle a câte A litri fiecare este nevoie pentru a umple un vas de B litri. Toate variabilele se vor citi de la tastatură.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> int main(){ int a,b,rez; // Introducem volumele printf("Introducem datele:\n"); scanf("%i %i",&a,&b); // Aplicarea operatorului rez=(b%a==0)?(b/a):((b/a)+1); printf("%i\n",rez); return 0; }</pre>	<pre>#include <iostream> using namespace std; int main(){ int a,b; // Introducem volumele cout<<"Introducem datele:\n"; cin>>a>>b; // a<b; // Aplicarea operatorului cout<<(b%a==0?b/a:(b/a)+1); return 0; }</pre>

Problema 2.3 - 06

Folosind operatorul condițional, elaborați un cod C/C++ care va determina soluțiile unei ecuații de gradul II.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> #include<stdbool.h> #include <math.h> int main(){ float a,b,c,d; // Introducem coeficientii printf("Care sunt laturile?"); scanf("%f %f %f",&a,&b,&c); d=pow(b,2)-4*a*c; float rd=sqrt(d); // Aplicarea operatorului</pre>	<pre>#include <iostream> #include <cmath> using namespace std; int main(){ int a,b,c; float d; // Introducem coeficientii cout<<"Introducem datele:"; cin>>a>>b>>c; d=pow(b,2)-4*a*c; float rd=sqrt(d); // Aplicarea operatorului</pre>

<pre>printf("Solutia este:\n"); d<0?printf("Nu are solutie!"): d==0?printf("%f",-b/(2.0*a)): printf("%f",-b-rd/(2*a)); printf(" %f",(-b+rd)/(2*a)); return 0; }</pre>	<pre>cout<<"Solutia este:\n"; d<0?cout<<"Nu are solutie!": d==0?cout<< -b/(2.0*a): cout<<(-b-rd)/(2*a); cout<<" "<<(-b+rd)/(2*a); return 0; }</pre>
--	--

Problema 2.3 - 07

Folosind operatorul condițional, elaborați un cod C/C++ care va determina bursa de merit al unui elev în baza mediei semestriale M. Valoarea mediei elevului va fi introdusă de la tastatură.

Implementarea C	Implementarea C++
<pre>#include<stdio.h> int main(){ float m; // Introducem media printf("Introducem media:\n"); scanf("%f",&m); // Aplicarea operatorului printf("Solutia este:\n"); (m<5)?printf("Nu are bursa!"): (m>=5&& m<6)?printf("%f",m*60): (m>=6&& m<7)?printf("%f",m*70): (m>=7&& m<8)?printf("%f",m*80): (m>=8&& m<9)?printf("%f",m*90): // Afișarea cu 2 zecimale printf("%.2f",m*100); return 0; }</pre>	<pre>#include <iostream> using namespace std; int main(){ float m; // Introducem media cout<<"Introducem media:\n"; cin>>m; // Aplicarea operatorului cout<<"Solutia este:\n"; (m<5)?cout<<"Nu are bursa!": (m>=5 && m<6)?cout<<(m*60): (m>=6 && m<7)?cout<<(m*70): (m>=7 && m<8)?cout<<(m*80): (m>=8 && m<9)?cout<<(m*90): cout<<(m*100); return 0; }</pre>

Problema 2.3 - 08

Folosind operatorul condițional, elaborați un cod C/C++ care va determina dacă o persoană în vârstă de N ani are anumite interdicții cu privire la alcool, țigări sau permis de conduce în SUA conform legii în vigoare.

Implementarea C	Implementarea C++
<pre>#include <stdio.h> int main(){</pre>	<pre>#include <iostream> using namespace std;</pre>

<pre> int N; // Introducem varsta printf("Introducem varsta:\n"); scanf("%d",&N); // Aplicarea operatorului printf("\nSolutia este:\t"); (N>=16)? printf("\nPersoana poate conduce un automobil!"): (N>=21)? printf("\nPersoana are dreptul la consumul de alcool și tigări!"): printf("\nPersoana este consi- derata minora!"); return 0; } </pre>	<pre> int main(){ int N; // Introducem varsta cout<<"Introducem varsta:\n"; cin>>N; // Aplicarea operatorului cout<<"Solutia este:\n"; (N>=16)?cout<<"Persoana poate conduce un automobil!": (N>=21)?cout<<"Persoana are dreptul la consumul de alcool și tigări!": cout<<"Persoana este conside- rata minora!"; return 0; } </pre>
---	---

Problema 2.3 - 09

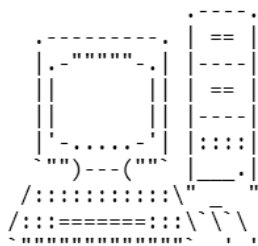
Folosind operatorul condițional, elaborați un cod C/C++ care va citi măsurile a două unghiuri exprimate în grade și minute. Programul afișează media celor două unghiuri, exprimată ca unghi, adică în grade, minute și secunde.

Implementarea C	Implementarea C++
<pre> #include <stdio.h> int main(){ int g1,m1,g2,m2,s; // Introducem datele printf("Introducem unghiul 1:"); scanf("%d %d",&g1,&m1); printf("Introducem unghiul 2:"); scanf("%d %d",&g2,&m2); // Aplicarea operatorului s=g1*60+m1+g2*60+m2; printf("Solutia este:\n"); printf("%d %d ",s/2/60,s/2%60); (s%2==0)?printf("%d",0): printf("%d",30); return 0; } </pre>	<pre> #include <iostream> using namespace std; int main(){ int g1,m1,g2,m2,s; // Introducem datele cout<<"Introducem unghiul 1:"; cin>>g1>>m1; cout<<endl; cout<<"Introducem unghiul 2:"; cin>>g2>>m2; cout<<endl; // Aplicarea operatorului s=g1*60+m1+g2*60+m2; cout<<"Solutia este:\n"; cout<<s/2/60<<" "<<s/2%60<<" "; (s%2==0)?cout<<0:cout<<30; return 0; } </pre>

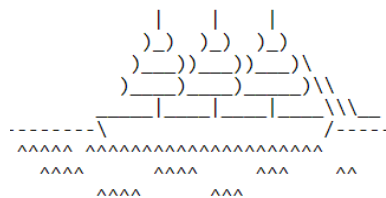
2.4 Probleme propuse

1. Probleme de aplicare a transformărilor elementare

1. *Elaborați un cod C/C++ care va afișa imaginea unui calculator de birou (A) sau a unei corăbii (B). Pentru a realiza un asemenea desen, vom aplica simbolurile textuale (codul ASCII). Modele de imagini pot fi cele de mai jos. Fiecare elev va aplica propria creativitate pentru prezentarea produsului final propriu.*



A



B

2. *Elaborați un cod C/C++ care va afișa la ecran imaginea unor figuri geometrice. Pentru a realiza un asemenea desen, vom aplica simbolurile textuale (codul ASCII). Un model de imagine poate fi cel din partea dreaptă. Fiecare elev va aplica simbolurile în mod creativ pentru prezentarea produsului final propriu.*



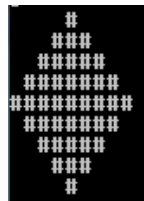
A



B



C



D

3. *Elaborați un cod C/C++ care va afișa numărul format din prima cifră de la primul număr, cifra a doua cea mai mare de la ambele numere și cifra a treia de la ultimul număr, dacă se introduc de la tastatură două numere care conțin câte trei cifre fiecare.*

4. *Elaborați un cod C/C++ care citește de la tastatură un număr natural cu trei cifre și afișează apoi numărul obținut prin eliminarea:*
 - a) primei cifre din număr;
 - c) cifrei din mijloc.
 - b) ultimei cifre din număr;
 - d) unei cifre oarecare.
5. *Elaborați un cod C/C++ care va afișa două numere naturale care au suma:*
 - a) 552, dacă cel mai mare împărțit la cel mai mic dă câtul 54 și restul 2;
 - b) 484, dacă cel mai mare împărțit la cel mai mic dă câtul 19 și restul 4;
6. *Elaborați un cod C/C++ care afișează două numere naturale care au diferența:*
 - a) 621, dacă cel mai mare împărțit la cel mic dă câtul 89 și restul 5;
 - b) 594, dacă cel mai mare împărțit la cel mic dă câtul 2 și restul 225;
7. *Elaborați un cod C/C++ care va afișa trei numere naturale care au suma:*
 - a) 60, dacă primul împărțit la al doilea dă câtul 10 și restul 3, iar primul împărțit la al treilea dă câtul 26 și restul 1.
 - b) 90, dacă primul împărțit la al doilea dă câtul 1 și restul 10, iar primul împărțit la al treilea dă câtul 1 și restul 5.
8. *Elaborați un cod C/C++ care va citi două valori întreaga pozitivă N și K , iar apoi va determina dacă:*
 - a) N este un pătrat perfect;
 - c) N este divizibil cu un număr K ;
 - b) N este un cub perfect;
 - d) N conține doar cifre impare.
9. *Elaborați un cod C/C++ care va determina suma și produsul primelor N numere consecutive, începând cu un număr X .*
10. *Elaborați un cod C/C++ care va determina câte portocale are fiecare elev, dacă se știe că Aurel are de 5 ori mai multe portocale decât Rebeca, iar Evelina are de 4 ori mai multe decât Rebeca. Împreună au 340 portocale.*
11. *Elaborați un cod C/C++ care va determina câte alune are fiecare copil, dacă se știe că George are cu 39 mai multe alune decât Pavel, iar Flaviu are cu 11 mai multe decât Pavel și împreună toți au 1241 alune.*
12. *Elaborați un cod în C/C++ care va afișa media aritmetică, media geometrică și media hiperbolică a trei numere reale: a , b și c .*
13. *Elaborați un cod în C/C++ care va determina suma și produsul cifrelor unui număr N care conține exact 5 cifre (sau exact K cifre).*

14. *Elaborați un cod în C/C++ care va determina suma pe care o va obține persoana la sfârșitul anului, dacă se cunoaște că ea avea inițial la bancă o sumă de bani S , iar în primele 6 luni ale anului, dobânda a fost $p\%$, apoi în următoarele 6 luni a fost $q\%$ din suma la care s-a adăugat și dobânda pe lunile anterioare.*
15. *Elaborați un cod în C/C++ care va afișa mesajul: „Da!”, dacă fracția a/b poate fi simplificată prin k , în caz contrar afișați mesajul: „Nu!”.*
16. *Ana a calculat suma numerelor naturale mai mici sau egale cu n , iar Andreea suma numerelor naturale mai mici sau egale cu m . Profesoara de mate a calculat apoi diferența celor două sume și a obținut rezultatul S . Pentru o valoare S dată, aflați perechile (n,m) , cu $n > m$, astfel încât profesoara de mate să obțină rezultatul S .*
17. ****Coffeeman vrea să împartă c caramelle la p prichindei astfel încât primul să ia k caramelle, al doilea $k+1$ caramelle, al treilea $k+2$ caramelle, și așa mai departe. De asemenea, Coffeeman vrea să rămână cu cât mai puține caramelle. Aflați cu câte caramelle rămâne Coffeeman ².*
18. ****Pe poarta unei fabrici ies în ordine n colete, fiecare având un volum cunoscut. Coletele sunt transportate folosind camioane. Toate camioanele au aceeași capacitate C , iar procedura este următoarea: fiecare colet scos din fabrică este imediat încărcat într-un camion, și nu este posibil ca la încărcare să fie mai mult de un camion. Determinați numărul minim de camioane necesar pentru a transporta cele n colete.*
19. ****Elena a rămas singură acasă și vrea să facă plăcinte. Pentru aceasta are nevoie de x grame de făină, y grame de zahăr, z ml de lapte, p oua și m kg de mere. Știind că prețul unui kg de făină este px , al unui kg de zahăr este py , litrul de lapte costă pz , kilogramul de mere costă pm și ouale sunt pp lei/buc., să se afle prețul plăcintei Elenei.*
20. ****Lui Alexandru îi place să circule cu autobuzul, deși câteodată poate să fie prea aglomerat. Ajutați-l pe Alexandru să afle niște statistici. Urmărind traseul autobuzului și cunoscând câte persoane urcă U și coboară C la fiecare stație, aflați numărul total de persoane care au urcat Ut , numărul total de persoane care au coborât Ct și numărul de persoane aflate în autobuz după terminarea traseului Pt . Inițial în autobuz erau P pasageri.*

2 Problemele care conțin simbolul „***” înainte de conținutul problemei, reprezintă un tip de problemă formulată într-un limbaj nestandard față de celelalte propuse în această lucrare.

2. Probleme de aplicare a formulelor matematice

- Elaborați un cod C/C++ care va calcula aria și perimetrul unui:
 - pătrat;
 - dreptunghi;
 - romb;
 - paralelogram;
 - triunghi isoscel;
 - triunghi echilateral;
 - triunghi oarecare;
 - pentagon regulat;
 - hexagon regulat;
 - heptagon regulat;
 - octagon regulat;
 - eneagon regulat.
- Elaborați un cod C/C++ care va afișa la ecran rezultatul expresiei matematice. În acest exemplu litera A, reprezintă numărul de ordine al elevului în registrul grupei. Afișați soluția pentru funcțiile de mai jos:
 - $f(A) = 2 \cdot \cos\left(\frac{3\pi}{A}\right) \cos\left(\frac{5\pi}{A}\right) - 1$
 - $g(A) = \frac{\ln(A)}{e^{A+1}} - \frac{\sqrt[3]{A+5}}{|1-\sqrt{A}|} + 2 \cdot A^3$
 - $h(A) = \frac{\sqrt{A}}{e^{A+1}} + \sqrt[5]{9-A} - A^7$
 - $i(A) = \left(\frac{\sqrt[3]{-A}}{e^A} + A^3\right)^{\frac{3}{2}} - 1$
- Elaborați un cod C/C++ care va calcula și va afișa valoarea distanței dintre două puncte din plan sau spațiu, dacă sunt cunoscute coordonatele acestora: $A(x_a, y_a)$ și $B(x_b, y_b)$ sau $A(x_a, y_a, z_a)$ și $B(x_b, y_b, z_b)$.
- Elaborați un cod C/C++ care va determina unghiul α (în grade), aria S și perimetrul P ale unui triunghi cu laturile: a, b și c. De asemenea determinați aria S a cercului înscris în acest triunghi.
- Elaborați un cod C/C++ care va afla latura a, aria S, înălțimea h_c , mediana m_b ale unui triunghi cu laturile b, c și unghiul dintre ele α (în grade). De asemenea determinați aria S_c a cercului circumscris triunghiului.
- Elaborați un cod C/C++ care va afla cateta b, ipotenuza c, perimetrul P, aria S, mediana m_c și bisectoarea b_c ale triunghiului dreptunghic cu cate-ta a și raza R a circumferinței circumscrise lui. De asemenea determinați diametrul d_i al circumferinței înscrise în acest triunghi.
- Elaborați un program C++ care va calcula și afișa Aria laterală, Aria bazei, Aria totală, perimetrul bazei și volumul paralelipipedului dreptunghic ABCDEFGH, unde avem: AB - lungime (notată L), BC - lățime (notată l), AF - înălțime (notată h).
- Elaborați un program C++ care va calcula și afișa Aria laterală, Aria bazei, Aria totală, perimetrul bazei (P_b) și volumul piramidei patrulatere re-

gulate $VABCD$, unde avem: AB - muchia bazei (notată m), VA - muchia laterală (notată l), VO - înălțimea piramidei (notată h), VM - apotema piramidei (notată a_p), OM - apotema bazei (notată a_b).

9. Elaborați un program C++ care va calcula și afișa Aria laterală, Aria bazei, Aria totală și volumul cilindrului $ABB'A'$, unde avem: AA' - generatoarea cilindrului (notată g), OO' - înălțimea cilindrului (notată h). Deoarece cilindrul este circular drept avem $g=h$, AO - raza bazei (notată r).
10. Elaborați un program C++ care va calcula și afișa Aria laterală, Aria bazei, Aria totală și volumul conului VAB , unde avem: VA - generatoarea conului (notată g), VO - înălțimea conului (notată h), AO - raza bazei (notată r).
11. Elaborați un program C++ care va calcula și afișa Aria laterală, Aria bazei mari, Aria bazei mici, Aria totală și volumul trunchiului de con în care avem: $A'A$ - generatoarea trunchiului de con (notată g), OO' - înălțimea trunchiului de con (notată h), AO - raza bazei mari (notată R), $A'O'$ - raza bazei mici (notată r).
12. Elaborați un program C++ care va afișa volumul V , aria totală A_t , aria laterală A_l și centrul de greutate GO , dacă se cunoaște raza R și înălțimea calotei sferice h .
13. Elaborați un program C++ care va afișa aria laterala (A_l), aria totală (A_t), volumul (V), lungimea rezei sferei înscrise în cub (r), dacă se știe că latura unui cub este notată cu a .
14. Elaborați un program C++ care va afișa aria bazei (A_b), perimetrul (P_b), aria laterala (A_l), aria totală (A_t) și volumul (V) al unei prisme triunghiulare regulate cu latura a .
15. Elaborați un program C++ care va afla latura a , înălțimea h și aria S ale unui triunghi echilateral, lungimile circumferințelor (L și l) și ariile cercului înscris S_i și circumscris S_o , dacă se știe că perimetrul triunghiului dat este P_i .
16. Elaborați un program C++ care va afla latura a , înălțimea h și aria S ale unui pentagon regulat, lungimile circumferințelor (L și l) și ariile cercului înscris S_i și circumscris S_o , dacă se știe că perimetrul pentagonului regulat dat este P_p .
17. Elaborați un program C++ care va afla latura a , înălțimea h și aria S ale unui hexagon regulat, lungimile circumferințelor (L și l) și ariile cercului înscris S_i și circumscris S_o , dacă se știe că perimetrul hexagonului regulat dat este P_h .

3. Probleme de aplicare a operatorului condițional

1. *Elaborați un cod C/C++ care va citi de la tastatură un număr întreg n și va verifică dacă acesta este:*
 - a) *pozitiv sau negativ;*
 - b) *par sau impar;*
 - c) *divizibil cu 3 sau divizibil cu 5;*
 - d) *conține 2 sau 3 cifre sau 4 cifre;*
2. *Elaborați un cod C/C++ care va determina apartenența unui număr întreg N la o mulțime. Litera A , reprezintă numărul de ordine al elevului în registrul grupei. Introduceți la început valoarea lui A , apoi afișați soluția. Rezultatul se va afișa sub forma „Da” sau „Nu”.*
 - a) $N \in [(-\infty, -2-A) \cup (-1, 1) \cup (2+A, +\infty)]$
 - b) $N \in [(-\infty, -17) \cup (9, +\infty)] \cap (-10, 25+A)$
3. *Elaborați un cod C/C++ care va citi de la tastatură trei numere a , b și c , apoi va realiza următoarele operațiuni:*
 - a) *afișarea valorii maxime (minime) dintre aceste numere;*
 - b) *determinarea existenței printre ele două numere pozitive (negative);*
 - c) *afișarea rezultatului, dacă suma (produsul) lor este un număr pozitiv par (impar);*
 - d) *afișarea rezultatului, dacă produsul (suma) lor este un număr negativ (pozitiv) impar;*
 - e) *afișarea rezultatului, dacă numerele sunt citite în ordine ascendentă (descendentă);*
 - f) *afișarea rezultatului, dacă numerele respectă condițiile teoremei lui Pitagora.*
4. *Elaborați un cod C/C++ care va demonstra următoarele legi ale lui De Morgan, iar rezultatul se va afișa sub forma „Adevărat” sau „Fals”, dacă se știe că p și q sunt două valori booleene.*
 - a) $!(p \ \&\& \ q) == !p \ || \ !q;$ - “Negația disjuncției este conjuncția negațiilor.”
 - b) $!(p \ || \ q) == !p \ \&\& \ !q.$ - “Negația conjuncției este disjuncția negațiilor.”
5. *Elaborați un cod C/C++ care va verifica un caracter introdus de la tastatură. Dacă este o literă mare, să afișeze mesajul “Literă mare”, dacă este*

literă mică să se afișeze mesajul “Literă mică”, altfel să se afișeze mesajul “Nu este literă”.

6. *Elaborați un cod C/C++ care va determina cel mai mare și cel mai mic număr care se poate forma din cifrele sale, dacă se știe că acest număr care conține exact trei cifre este introdus de la tastatură.*
7. *Elaborați un cod C/C++ care va determina dacă 3 numere întregi pot fi laturile unui triunghi isoscel și să se determine perimetrul și aria acestui triunghi.*
8. *Elaborați un cod C/C++ care va determină dacă 3 numere întregi pot fi laturile unui triunghi echilateral și să se determine raza cercului înscris și raza cercului circumscris acestui triunghi.*
9. *Elaborați un cod C/C++ care va determina dacă 3 numere pot fi unghiurile unui triunghi și să se determine tipul triunghiului (isoscel, echilateral, dreptunghic, dreptunghic isoscel sau scalen).*
10. *Elaborați un cod C/C++ care va determina vârsta unei persoane în ani impliniți, dacă se cunoaște data de naștere a unei persoane (anul, luna și ziua) și data curentă exprimată la fel.*
11. *Elaborați un cod C/C++ care va determina ce zi a săptămânii a fost, dacă se cunoaște data de naștere a unei persoane (anul, luna și ziua).*
12. *Elaborați un cod C/C++ care va verifica dacă numerele reale a și b, respectiv c și d formează intervale. Dacă formează intervale, să se verifice dacă acestea sunt disjuncte.*
13. *Elaborați un cod C/C++ care va afișa lungimea segmentului mai mare, dacă se citesc de la tastatură coordonatele carteziane ale extremităților a două segmente.*
14. *Elaborați un cod C/C++ care va stabili dacă se pot așeza pe un balansoar 3 copii care au greutatea a, b și c, astfel încât acesta să stea în echilibru.*
15. ****Andrian este trimis de mama sa la cumpărături și are de plătit o sumă de S lei. El are în buzunar bancnote cu valorile a, b și c. Afișați câte bancnote de fiecare fel folosește, astfel încât să dea vânzătorului cât mai puține bancnote pentru a plăti integral suma S. Se presupune că Andrian are suficiente bancnote de fiecare fel. Datele de intrare asigură că suma poate fi plătită. Rezultatele se vor afișa ca în exemplul de mai jos, adică nu se vor afișa tipurile de bancnote care nu sunt folosite.*

Exemplu:

- $S=75, a=50, b=30, c=5 \Rightarrow$ se va afișa $1*50 \ 5*5$;
- $S=170, a=100, b=20, c=10 \Rightarrow$ se va afișa $1*100 \ 3*20 \ 1*10$.

APLICAȚII ALE INSTRUCȚIUNILOR DECIZIONALE ÎN C/C++

3.1 Aplicarea instrucțiunii IF



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit instrucțiunea IF, unele probleme au fost deja rezolvate cu ajutorul operatorului ternar (condițional) în capitolul precedent.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 3.1 - 01

Se citesc de la tastatură două numere naturale a și b cu exact două cifre fiecare. Scrieti programul C++ care afișează numărul total de cifre pare din cele două numere dacă ele au aceeași paritate, iar dacă au parități diferite, atunci afișează numărul total de cifre impare din cele două numere.

Exemple:

- Dacă $a = 78$ și $b = 18$ se va afișa 2 (8 și 8 sunt cifrele pare, adică 2).
- Dacă $a = 36$ și $b = 55$ se va afișa 3 (sunt 3 cifre impare).

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int a,b;
    cout<<"Introducem numerele:"; cin>>a>>b;
    if(a%2==b%2)
        cout<<(a/10%2==0)+(a%2==0)+(b/10%2==0)+(b%2==0);
```

```

else cout<<a/10%2+a%2+b/10%2+b%2;
return 0;
}

```

Problema 3.1 - 02

Se citește un număr natural n . Să se descompună ca produs de două numere naturale consecutive. Dacă acest lucru nu este posibil, atunci să se afișeze mesajul "IMPOSIBIL".

Exemple:

- 10 se va afișa IMPOSIBIL;
- $420 = 20 \cdot 21$.

Implementarea C++

```

#include <iostream>
#include <cmath> // Biblioteca ce operează cu funcții matematice
using namespace std;
int main(){
    unsigned int n,x,d;
    cout<<"Introducem valoarea lui n:"; cin>>n;
    //Solutia se obtine din rezolvarea ecuatiei x*x+x-n=0
    d=1+4*n;
    if(sqrt(d)==floor(sqrt(d))){
        //floor(a): trunchiază pe a la cel mai apropiat număr întreg mai
        mic decât el și returnează rezultatul.
        x=(-1+sqrt(d))/2; cout<<x<<" "<<x+1;
    }
    else cout<<"IMPOSIBIL"; return 0;
}

```

Problema 3.1 - 03

Se citesc 3 numere naturale z , l și a , reprezentând o dată calendaristică în format: zi lună an. Să se afișeze data zilei anterioare.

Exemple:

- 1 2 2000 => 31 1 2000;
- 1 3 2000 => 29 2 2000.

Implementarea C++

```

#include <iostream>
using namespace std;
int main(){
    int z,l,a;

```

```
cout<<"Introducem datele:"; cin>>z>>l>>a;
if(z>1) z--;
else{
    l--;
    if(l==0) {z=31;l=12; a--;}
    else if(l==1 || l==3 || l==5 || l==7 || l==8 || l==10)
        z=31; // luna are 31 de zile
    else if(l==4 || l==6 || l==9 || l==11)
        z=30; // luna are 30 de zile
    else if(a%4==0 && a%100!=0 || a%400==0) z=29;
    else z=28; // luna are 28 de zile
}
cout<<"Solutia problemei este: "<<z<<" "<<l<<" "<<a;
return 0;
}
```

Problema 3.1 - 04

Se citesc două numere naturale *a* și *l*, unde *a* reprezintă un an, iar *l* este numărul unei luni din anul *a*. Afișați câte zile are luna *l*. Se va ține cont dacă anul *a* este bisect.

Exemplu:

- Pentru *a*=2008 și *l*=2, rezultatul este 29, deoarece 2008 este an bisect.

Implementarea C++
<pre>#include<iostream> using namespace std; int main(){ int a,l; cout<<"Introducem anul si luna: "; cin>>a>>l; cout<<"\nSolutia problemei este: \n"; cout<<"\tLuna a "<<l<<"-a are "; if(l==1 l==3 l==5 l==7 l==8 l==10 l==12) cout<<31; // luna are 31 de zile else if(l==4 l==6 l==9 l==11) cout<<30; else if(a%4==0 && a%100!=0 a%400==0) cout<<29; else cout<<28; // luna are 28 de zile cout<<" zile in anul "<<a<<endl; return 0; }</pre>

Problema 3.1 - 05

Se citesc 3 numere naturale a , b și c . Să se afișeze în ordine ascendentă a valorii lor. Rezolvați problema prin cel puțin două metode.

Exemplu:

- Pentru $a=28$, $b=12$ și $c=21$, rezultatul este: $b\ c\ a$, adică 12 21 28.

Implementarea C++

```
#include<iostream>
using namespace std;
int main(){
    int a,b,c,aux;
    cout<<"Introducem numerele: "; cin>>a>>b>>c;
    // Vom aplica operatorul logic "&"
    cout<<"\nMetoda de rezolvare 1:\n\t";
    if(a<=b && b<=c) cout<<a<<" "<<b<<" "<<c;
        else if(a<=c && c<=b) cout<<a<<" "<<c<<" "<<b;
        else if(b<=a && a<=c) cout<<b<<" "<<a<<" "<<c;
        else if(b<=c && c<=a) cout<<b<<" "<<c<<" "<<a;
        else if(c<=a && a<=b) cout<<c<<" "<<a<<" "<<b;
        else if(c<=b && b<=a) cout<<c<<" "<<b<<" "<<a;
    // Vom aplica o variabilă auxiliară
    cout<<"\nMetoda de rezolvare 2:\n\t";
    if(a>b) { aux=a; a=b; b=aux; }
    if(a>c) { aux=a; a=c; c=aux; }
    if(b>c) { aux=b; b=c; c=aux; }
    cout<<a<<" "<<b<<" "<<c;
    return 0;
}
```

Problema 3.1 - 06

Se citesc 3 numere întregi a , b și c . Să se afișeze cel mai mare dintre ele (elementul maximal). Rezolvați problema prin cel puțin două metode.

Exemplu:

- Pentru $a=28$, $b=12$ și $c=21$, rezultatul este: a , adică 28.

Implementarea C++

```
#include<iostream>
using namespace std;
int main(){
    int a,b,c,maxim;
```



```
cout<<"Introducem numerele: "; cin>>a>>b>>c;
// Vom aplica operatorul logic "&"
cout<<"\nMetoda de rezolvare 1:\n\t";
if(a>=b && a>=c) cout<<a;
else if(b>=a && b>=c) cout<<b;
else if(c>=a && c>=b) cout<<c;
// Vom aplica o variabilă auxiliară
cout<<"\nMetoda de rezolvare 2:\n\t";
maxim=a;
    if(b>maxim) maxim=b;
    if(c>maxim) maxim=c;
    cout<<maxim;
return 0;
}
```

Problema 3.1 - 07

Se citesc două numere întregi a și b. Dacă a divide pe b sau b divide pe a, atunci să se afișeze câtul împărțirii celui mai mare la cel mai mic, iar altfel restul împărțirii celui mai mare la cel mai mic.

Implementarea C++
<pre>#include<iostream> using namespace std; int main(){ int a,b; cout<<"Introducem numerele: "; cin>>a>>b; if(a%b==0 b%a==0){ if(a>b) cout<<a/b; else cout<<b/a; } else if(a>b) cout<<a%b; else cout<<b%a; return 0; }</pre>

Problema 3.1 - 08

Se citesc două numere întregi P și V, care reprezintă prețul unui produs și respectiv prețul cu care a fost vândut un produs. Determinați profitul sau pierderea. Toate datele se vor introduce de la tastatură.

Implementarea C++

```
#include<iostream>
using namespace std;
int main(){
    int P,V, suma;
    /* Prețul de intrare și prețul de vânzare al unui produs*/
    cout<<"Introducem numerele: "; cin>>P>>V;
    if(V > P){ /* Calculăm profitul */
        suma = V - P; cout<<"\tProfit = "<<suma;
    }
    else if(P > V){ /* Calculăm pierderea */
        suma = P - V; cout<<"\tPierdere = "<<suma;
    }
    else{ cout<<"\tLipsa profit! Lipsa pierdere!"; }
    return 0;
}
```

Problema 3.1 - 09

Se citește un număr real pozitiv E , care reprezintă numărul de Kw utilizați. Calculați factura de energie electrică în funcție de condiția dată: pentru primele 50 de unități avem 0,50/unitate, pentru următoarele 100 de unități avem 0,75/unitate, pentru următoarele 100 de unități avem 1,20/unitate, dacă avem peste 250, atunci va fi 1,50/unitate. La factură se adaugă o suprataxă suplimentară de 20%.

Implementarea C++

```
#include<iostream>
using namespace std;
int main(){
    float E, S, Total, St;
    // Unitatea de intrare consumată de la utilizator
    cout<<"Introducem datele: "; cin>>E;
    // Calculăm factura electricității în funcție de condiții
    if(E <= 50){ S = E * 0.50; }
    else if(E <= 150){ S = 25 + ((E-50) * 0.75); }
    else if(E <= 250){ S = 100 + ((E-150) * 1.20); }
    else { S = 220 + ((E-250) * 1.50); }
    //Calculăm factura totală a energiei electrice cu suprataxă
    St = S * 0.20; Total = S + St;
    cout<<"\nFactura de energie electrica: "<<Total;
}
```

3.2 Aplicarea instrucțiunii SWITCH



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit comutatorul, instrucțiunea SWITCH.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 3.2 - 01

Elaborați un program C/C++ care va permite introducerea numărului zilei N , cu valori din intervalul 1-7 și va imprima numele zilei săptămânii folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă $N=1$ se va afișa „Luni”;
- Dacă $N=2$ se va afișa „Marți”;
- Dacă $N=4$ se va afișa „Joi”;
- Dacă $N=5$ se va afișa „Vineri”.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int zi;
    cout<<"Introducem numarul zilei: "; cin>>zi;
    cout<<"Ati ales ziua de: ";
    switch(zi){
        case 1: cout<<"Luni"; break;
        case 2: cout<<"Marti"; break;
        case 3: cout<<"Miercuri"; break;
        case 4: cout<<"Joi"; break;
        case 5: cout<<"Vineri"; break;
        case 6: cout<<"Sambata"; break;
        case 7: cout<<"Duminica"; break;
        default: cout<<"Introduceti numarul zilei intre 1-7.";
    }
}
```

Orice problemă poate fi reformulată, spre exemplu, la introducerea numărului zilei 6 și 7 să se afișeze mesajul „Week-end”.

Problema 3.2 - 02

Elaborați un program C/C++ care va permite introducerea numărului lunii N, cu valori din intervalul 1-12 și va imprima numărul de zile corespunzătoare lunii folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă N = 1 se va afișa „31 zile”;
- Dacă N = 4 se va afișa „30 zile”.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int luna;
    cout<<"Introducem numarul lunii: "; cin>>luna;
    cout<<"Ati ales ziua de: ";
    switch(luna){
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            cout<<"31 de zile"; break;
        case 4: case 6: case 9: case 11: cout<<"30 de zile"; break;
        case 2: cout<<"28 (29 de zile in anul bisect)"; break;
        default: cout<<"Introduceti numarul lunii intre 1-12.";
    }
    return 0;
}
```

Problema 3.2 - 03

Elaborați un program C/C++ care va permite introducerea unui caracter C, care reprezintă o literă din alfabetul englez și va imprima mesajul: „Este o vocală” sau „Este o consoană” folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă C = 'a' se va afișa „Este o vocală”;
- Dacă C = '5' se va afișa „Este o cifră de la 0 la 9”;
- Dacă C = 'b' se va afișa „Este o consoană sau alt simbol”.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    char c;
    cout<<"Introducem litera: "; cin>>c;
```

```

    cout<<"\nCaracterul introdus: ";
    switch(c){
        case 'a': case 'e': case 'i': case 'o': case 'u':
            cout<<"Este o vocala."; break;
        case 'A': case 'E': case 'I': case 'O': case 'U':
            cout<<"Este o vocala."; break;
        case '0': case '1': case '2': case '3': case '4':
        case '5': case '6': case '7': case '8': case '9':
            cout<<"Este o cifra de la 0 la 9"; break;
        default: cout<<"Ati introdus o consoana sau alt simbol.";
    }
    return 0;
}

```

Problema 3.2 - 04

Elaborați un program C/C++ care va permite introducerea a două numere întregi A și B și va imprima elementul maxim și cel minim folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă A = 10 și B = 15, se va afișa: „Maxim=15” și „Minim=10”;
- Dacă A = 25 și B = 15, se va afișa: „Maxim=25” și „Minim=15”.

Implementarea C++

```

#include <iostream>
using namespace std;
int main(){
    int a,b;
    cout<<"Introducem numerele: "; cin>>a>>b;
    cout<<endl;
    switch(a>b){
        case 0: cout<<"\tMaximal este "<<b; break;
        case 1: cout<<"\tMaximal este "<<a; break;
    }
    cout<<endl;
    switch(a<b){
        case 0: cout<<"\tMinimal este "<<b; break;
        case 1: cout<<"\tMinimal este "<<a; break;
    }
    return 0;
}

```

Problema 3.2 - 05

Elaborați un program C/C++ care va permite introducerea unui număr întreg A și va imprima mesajul: „Este par” sau „Este impar” folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă $A = 10$, se va afișa: „Este par”;
- Dacă $A = 25$, se va afișa: „Este impar”.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int a;
    cout<<"Introducem numarul: "; cin>>a;
    cout<<endl;
    switch(a%2){
        case 0: cout<<"\tEste par "; break;
        case 1: cout<<"\tEste impar "; break;
    }
    return 0;
}
```

Problema 3.2 - 06

Elaborați un program C/C++ care va permite introducerea unui număr întreg A și va imprima mesajul: „Este pozitiv” sau „Este negativ” sau „Este zero” folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă $A = 10$, se va afișa: „Este pozitiv”;
- Dacă $A = -25$, se va afișa: „Este negativ”;
- Dacă $A = 0$, se va afișa: „Este zero”.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int a;
    cout<<"Introducem numarul: "; cin>>a;
    cout<<endl;
    switch(a>0){
```

```

        case 1: cout<<"\tEste pozitiv "; break;
        case 0:
            switch(a<0){
                case 1: cout<<"\tEste negativ "; break;
                case 0: cout<<"\tEste zero "; break;
            }
            break;
    }
    return 0;
}

```

Problema 3.2 - 07

Elaborați un program C/C++ care va permite introducerea a trei numere reale A , B și C , care reprezintă coeficienții ecuației pătratice și va imprimă soluțiile acestora folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă $A = 4$, $B = -2$ și $C = 10$, se va afișa: $x_1=1.85$ și $x_2=1.35$;
- Dacă $A = 2$, $B = 5$ și $C = 9$, se va afișa: $x_1=0.46$ și $x_2=0.46$.

Implementarea C++

```

#include <iostream>
#include <cmath>
using namespace std;
int main(){
    float a,b,c;
    float x1,x2,i,delta;
    cout<<"Introducem coeficientii: "; cin>>a>>b>>c;
    cout<<endl;
    delta = pow(b,2)-(4*a*c);
    switch(delta>0){
        case 1:
            x1 = (-b+sqrt(delta))/(2*a);
            x2 = (-b-sqrt(delta))/(2*a);
            cout<<"Solutiile reale sunt: \n\t";
            cout<<"x1= "<<x1<<" & x2= "<<x2<<endl; break;
        case 0:
            switch(delta<0){
                case 1:
                    x1 = x2 = -b/(2*a);
                    i = sqrt(-delta)/(2*a);
                    cout<<"Solutiile complexe sunt: \n\t";
                    cout<<"x1= "<<x1+i<<" & x2= "<<x2+i<<endl;

```

```

        break;
    case 0:
        x1 = x2 = -b/(2*a);
        cout<<"Solutiile complexe sunt: \n\t";
        cout<<"x1= "<<x1<<" & x2= "<<x2<<endl; break;
    }
}
return 0;
}

```

Problema 3.2 - 08

Elaborați un program C/C++ care va permite crearea unui meniu vertical din care se vor selecta operațiile aritmetice fundamentale. Este necesar de a introduce două numere reale A și B, imprimați soluția obținută în dependență de operatorul selectat folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă $A = 4$, $B = 6$ și operația = '+', se va afișa: „4 + 6 = 10”;
- Dacă $A = 5.2$, $B = 3$ și operația = '-', se va afișa: „5.2 - 3 = 2.2”.

Implementarea C++

```

#include <iostream>
using namespace std;
int main(){
    char op; float a, b, result;
    cout<<"Introducem numerele: "; cin>>a>>b;
    cout<<"Meniul de operatori:\n";
    cout<<"\t+ => "<<a<<" + "<<b<<"\n";
    cout<<"\t- => "<<a<<" - "<<b<<"\n";
    cout<<"\t* => "<<a<<" * "<<b<<"\n";
    cout<<"\t/ => "<<a<<" / "<<b<<"\n";
    cout<<"Introducem operatie: "; cin>>op; cout<<endl;
    switch(op){
        case '+': result = a + b; break;
        case '-': result = a - b; break;
        case '*': result = a * b; break;
        case '/': result = a / b; break;
        default: cout<<"Operatorul nu corespunde!";
    }
    cout<<"\t"<<a<<" "<<op<<" "<<b<<" = "<<result<<endl;
    return 0;
}

```


Problema 3.2 - 09

Elaborați un program C/C++ care va permite generarea unui număr întreg N din intervalul $[0, 49]$ și va afișa textul color corespunzător celor 5 criterii de divizare a acestui interval. Imprimați soluția obținută în dependență de numărul generat folosind comutatorul, instrucțiunea SWITCH.

Exemple:

- Dacă $N = 4$, atunci se va afișa textul din consolă de culoare verde;
- Dacă $N = 30$, atunci se va afișa textul din consolă de culoare violet.

Implementarea C++

```
#include <iostream>
// Fișierul antet <stdlib.h> definește mai multe funcții cu scop general, inclusiv gestionarea dinamică a memoriei, generarea aleatorie de numere, comunicarea cu mediul, aritmetica întregului, căutarea, sortarea și conversia.
#include <stdlib.h>
// Fișierul antet <ctime> declară un set de funcții, macrocomenzi și tipuri pentru a lucra cu data și ora.
#include <ctime>
using namespace std;
int main(){
// Funcția srand() stabilește punctul de plecare pentru producerea unei serii de numere întregi pseudo-aleatorii.
    srand((unsigned) time(0));
// Funcția rand() este utilizată pentru a genera numere aleatorii în intervalul [0, RAND_MAX).
    int n = 1+(rand()%50);
    cout<<"Numarul generat este: "<<n<<endl;
    switch(n){
        case 0 ... 9:
            system("Color 0A");cout<<"Verde"; break;
        case 10 ... 19:
            system("Color 0B");cout<<"Cyan"; break;
        case 20 ... 29:
            system("Color 0C");cout<<"Rosu"; break;
        case 30 ... 39:
            system("Color 0D");cout<<"Violet"; break;
        case 40 ... 49:
            system("Color 06");cout<<"Galben"; break;
    }
}
```

3.3 Probleme propuse

1. Probleme de aplicare a instrucțiunii IF

1. *Elaborați un program în C/C++ care citește valoarea variabilei X și afișează valoarea funcției $f(x)$.*

$$\text{a) } f(x) = \begin{cases} 3 \cdot x^2 - 7 \cdot x - 10 & | x < 0 \\ 2 & | x = 0 \\ 4 \cdot x^2 & | x > 0 \end{cases}$$

$$\text{b) } f(x) = \begin{cases} x - 3 & | x < 5 \\ x + 1 & | 5 < x < 25 \\ x^2 - 5 \cdot x + 6 & | x > 25 \end{cases}$$

2. *Elaborați un program în C/C++ care citește valorile variabilelor a , b și c ; și afișează valoarea expresiei E .*

$$\text{a) } E = \begin{cases} \sqrt{a+b} & | c > 0 \\ a \cdot b & | x = 0 \\ e^{a+b} & | c < 0 \end{cases}$$

$$\text{b) } E = \begin{cases} \sqrt{b^a} & | c > 0 \\ \ln(a) \cdot b & | c = 0 \\ a - b & | c < 0 \end{cases}$$

$$\text{c) } E = \begin{cases} a - b & | c > 0 \\ \ln(a) \cdot b & | x = 0 \\ \log(a+b) & | c < 0 \end{cases}$$

3. *Elaborați un program în C/C++ care citind de la tastatură o valoare întreagă N , va afișa pe ecran cele mai apropiate (din față și din spate):*

a) două numere pare;

c) două pătrate perfecte;

b) două numere impare;

d) două cuburi perfecte;

4. *Elaborați un program în C/C++ care citește valorile variabilelor a , b și c ; iar ca rezultat afișează diferența dintre elementul maximal și cel minimal.*

5. *Elaborați un program în C/C++ care citind de la tastatură o valoare X , unde $X \leq 365$. Afișați pe ecran în ce lună a anului se află ziua cu numărul X a anului.*

6. *Elaborați un program în C/C++ care afișează modelul și culoarea unui automobil în dependență de numărul de locuri din parcare. Se cunoaște că parcare include 10 locuri, dintre care 3 locuri sunt destinate pentru automobilele conduse de șoferi cu cerințe speciale, aceste locuri vor fi libere. Determinați care sunt aceste 3 locuri, dacă se știe că numărul poziției trebuie să fie divizibil cu 3.*

7. *Elaborați un program în C/C++ care citește valoarea variabilei a , unde a este un număr întreg pozitiv și afișează umărul respectiv cu cifre romane. Se cunoaște că: $I=1$, $V=5$, $X=10$, $L=50$, $C=100$, $D=500$, $M=1000$.*

8. Elaborați un program în C/C++ care citește valorile variabilelor t_1 și t_2 , unde t_1 și t_2 sunt numere întreg ce indică valori ale temperaturii în $^{\circ}\text{C}$. Afișați temperatura minimă și mesajul: „Este frig!”, dacă temperatura minimă este mai mică decât 4°C .
9. Elaborați un program în C/C++ care citește valorile variabilelor a , b și c , unde a , b și c sunt numere reale. Afișați în același program următoarele date: media aritmetică, media geometrică, media armonică și media hiperbolică.
10. Elaborați un program în C/C++ care citește valoarea variabilei a , unde a este un număr întreg pozitiv și determină dacă n este un pătrat sau cub perfect.
11. Elaborați un program în C/C++ care citește valorile variabilelor a și b , unde a și b sunt numere naturale nenule. Determinați dacă fracția $\frac{a}{b}$ poate fi simplificată prin numărul natural c . În caz afirmativ, să se afișeze fracția simplificată.
12. Elaborați un program în C/C++ care citește valoarea variabilei a , unde a este raza cercului construit cu centrul în originea sistemului de coordonate. Determinați apartenența unui punct $A(x,y)$ la cercul dat, unde x și y sunt coordonatele punctului în sistemul cartezian de coordonate. Stabiliți dacă punctul se află: pe o latură, în interiorul sau în exteriorul figurii.
13. Elaborați un program în C/C++ care citește coordonatele a trei puncte $A(x_1,y_1)$, $B(x_2,y_2)$ și $C(x_3,y_3)$, unde A , B și C reprezintă punctele ce formează un triunghi în sistemul cartezian de coordonate. Determinați apartenența la acest triunghi a unui punct $M(x,y)$, unde x și y sunt coordonatele punctului în sistemul cartezian de coordonate. Stabiliți dacă punctul se află: pe o latură, în interiorul sau în exteriorul figurii.
14. Elaborați un program în C/C++ care citește patru numere a , b , c și d . Considerând un sistem de axe ortogonale xOy , determinați poziția dreptelor $y_1=ax+b$ și $y_2=cx+d$. În caz că dreptele se intersectează, afișați și coordonatele punctului de intersecție.
15. ***Conform calendarului japonez, fiecare an poartă numele unui animal. Fiecare denumire se repetă exact o dată la 12 ani. Deci, un ciclu este format din 12 ani cu următoarele denumiri de animale în această ordine: șobolan, bou, tigr, iepure, dragon, șarpe, cal, oaie, maimuță, cocoș, ciine, porc. Știind că anul 2000 a fost anul Dragonului, să se scrie un algoritm care va citi de la tastatură anul (număr de patru cifre) și va afișa denumirea lui conform calendarului japonez.

2. Probleme de aplicare a instrucțiunii SWITCH



- ✓ Instrucțiunea IF și SWITCH reprezintă instrucțiuni de decizie, de aceea toate problemele propuse spre realizare prin intermediul instrucțiunii IF, se pot implementa și prin intermediul instrucțiunii SWITCH.
- ✓ Instrucțiunea IF este foarte asemănătoare cu operatorul ternar (condițional), de aceea toate problemele propuse spre realizare prin intermediul operatorului condițional, se pot implementa și prin intermediul instrucțiunii SWITCH.

1. Elaborați un program în C/C++ care citește valoarea variabilei N , unde N este un număr natural. Determinați ultima cifră a numărului 7^N .
2. Elaborați un program în C/C++ care citește valorile variabilelor ID și P , unde ID este un cod de identificare, iar P este un număr parolă. Când introducem ID -ul corect, atunci ne cere parola, iar când introducem parola corectă, se imprimă mesajul: „Bun venit, drag programator.”
3. Elaborați un program în C/C++ care citește valoarea variabilei a , unde $a > 1$. Să se rezolve în N (mulțimea numerelor naturale) ecuația $5x + 3y = a$. De exemplu, pentru $a = 8$ soluția va fi (1, 1).
4. Elaborați un program în C/C++ care citește valorile variabilelor a , b și c , ce reprezintă numere naturale diferite de zero. Să se rezolve în N (mulțimea numerelor naturale) ecuația $a^2 + b^2 = c^2$. De exemplu, pentru $c = 5$ soluția va fi (3, 4).
5. Elaborați un program în C/C++ care citește coordonatele carteziene ale trei vârfuri ale unui dreptunghi. Să se determine coordonatele celui de-al patrulea vârf. De exemplu, pentru coordonatele date: $A(5;1)$, $B(1;1)$ și $C(1;7)$, se va afișa punctul $D(5;7)$.
6. Elaborați un program în C/C++ care citește coordonatele carteziene ale trei vârfuri ale unui trapez isoscel. Să se determine coordonatele celui de-al patrulea vârf. De exemplu, pentru coordonatele date: $A(4;4)$, $B(2;1)$ și $C(9;1)$, se va afișa punctul $D(7;4)$.
7. Elaborați un program în C/C++ care verifică dacă intervalul $[a; b]$, respectiv $[c; d]$ se intersectează. Dacă intervalele se intersectează, să se afișeze

intervalul obținut în intersecție, dacă se știe că cele patru numere întregi: a, b, c și d se vor introduce de la tastatură.

8. Elaborați un program în C/C++ care verifică dacă intervalul $I_1: [a; b]$, respectiv $I_2: [c; d]$ se intersectează. Afișați intervalul obținut în urma diferenței dintre aceste două intervale de elemente, adică: $I_1 - I_2$ și $I_2 - I_1$, dacă se știe că cele patru numere întregi: a, b, c și d se introduc de la tastatură.
9. ***Agenția „HAPPY Holiday Tour” va oferi un tarif special pentru o vacanță în Insula Langkawi, Insula Pangkor și Insula Tioman în timpul vacanței școlare. Costul complet pentru 3 zile pentru un adult este următorul:

	Langkawi	Pangkor	Tioman
Zbor (2 sensuri)	120,00 \$	90,00 \$	150,00 \$
Hotel (3 zile)	260,00 \$	330,00 %	390,00 \$

Pentru o familie, reducerea se acordă în funcție de următoarele:

Odae cu pat (pentru 3 zile)	• 1 adult - nu se oferă reducere; • 2 adulți - 20% reducere fiecărui adult; • 3 sau mai mulți adulți - 30% reducere fiecărui adult; • 50% reducere fiecărui copil.
Zbor (2 sensuri)	• Fără reducere pentru adulți; • 50% reducere fiecărui copil.

Să presupunem că datele de intrare sunt: destinația, numărul total de adulți și numărul total de copii, determinați care este:

- reducerea oferită pentru copii;
- reducerea oferită pentru adulți;
- suma necesară de bani pentru ca familia să poată pleca în vacanță.

APLICAȚII ALE INSTRUCȚIUNILOR CICLICE ÎN C/C++

4.1 Aplicarea instrucțiunii FOR



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit instrucțiunea ciclică FOR.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 4.1 - 01

Elaborați un program C/C++ care va permite calcularea sumei primelor n numere naturale.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int N,i,s=0;
    cout<<"Introducem valoarea lui N: "; cin>>N;
    for(i=1;i<=N;i++){
        s+=i; p*=i;    //calculăm suma (s) și produsul (p);
    }
    cout<<"Suma primelor "<<N<<" numere naturale: \t"<<s<<endl;
}
```

Problema 4.1 - 02

Elaborați un program C/C++ care va permite calcularea valorii expresiei: $S=1+(1\cdot 2)+(1\cdot 2\cdot 3)+\dots+(1\cdot 2\cdot \dots \cdot n)$.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int N,i,s=0,p=1;
    cout<<"Introducem valoarea lui N: \t"; cin>>N;
    for(i=1;i<=N;i++){
        p*=i;    //calculăm produsul
        s+=p;    //calculăm suma
    }
    cout<<"Rezultatul expresiei este: \t";
    cout<<s<<endl;
}
```

Problema 4.1 - 03

Elaborați un program C/C++ care va permite calcularea valorii expresiei: $P=2 \cdot 4 \cdot 6 \cdot 8 \cdot \dots \cdot n$, unde $n=2 \cdot k$.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int n,i,p=1;
    cout<<"Introducem valoarea lui N: \t"; cin>>n;
    n*=2;
    for(i=2;i<=n;i++){
        if (!(i%2)) p*=i;
        //se mai putea scrie if(i%2==0) <=> testeaza dacă e par
    }
    cout<<"Rezultatul expresiei este: \t"<<p<<endl;
}
```

Problema 4.1 - 04

Elaborați un program C/C++ care va permite citirea de la tastatură a unui număr natural P, care este perimetrul unui triunghi isoscel. Afișați toate tripletele de numere naturale ce pot reprezenta lungimile laturilor acestui triunghi.

Exemplu:

- Dacă $P = 12$, atunci avem tripletele: (4, 4, 4) și (5, 5, 2);

Implementarea C++

```
#include<iostream>
using namespace std;
int main(){
    int a,c,p,k=0;
    cout<<"Introducem perimetrul: \t"; cin>>p;
    cout<<"Tripletele sunt: \n";
    for(a=1;a<=p/2;a++){
        for(c=1;c<=p/2;c++){
            if(2*a+c==p && 2*a>c){
                k++;
                cout<<"\t"<<a<<" "<<a<<" "<<c<<endl;
            }
        }
    }
    cout<<"Numarul total de triplete: "<<k;
}
```

Problema 4.1 - 05

Elaborați un program C/C++ care va permite transformarea unui număr N din baza 10 într-o bază B , unde $2 \leq B < 10$.

Exemple:

- Dacă $N = 125_{10}$ și $B = 2$, atunci se va afișa $1\ 111\ 101_2$;
- Dacă $N = 125_{10}$ și $B = 7$, atunci se va afișa 236_7 ;

Implementarea C++

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
    int N,NBi,rez,B;
    cout<<"Introducem un numar in baza 10: \t"; cin>>N;
    cout<<"Introducem un numar ca baza: \t"; cin>>B;
    // Un alt mod de a scrie o instructiune for
    int k=N; // copiem valoarea variabilei N in variabila k
    for (NBi=0; N; NBi=10*NBi+N%B+1,N=N/B);
    for (rez=0; NBi; rez=10*rez+(NBi-1)%10,NBi=NBi/10);
    system("CLS"); // instructiunea de curățare a ecranului
    cout<<"\nNumarul "<<k<<" in baza "<<B<<" este: \t"<<rez;
}
```


Problema 4.1 - 06

Elaborați un program C/C++ care va permite citirea unui număr natural N și afișează N pătrate ca în exemplul alăturat.

Exemplu:

Pentru n=2 avem:

```
1
2 2
2 2
```

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int N,i,x,y;
    cout<<"Introducem N: \t"; cin>>N;
    for(i=1;i<=N;i++){
        for(x=1;x<=i;x++){
            for(y=1;y<=i;y++) cout<<i<<" "; cout<<endl;
        }
    }
}
```

Problema 4.1 - 07

Elaborați un program C/C++ care va permite citirea unui număr natural N și afișează un triunghi de numere asemenea cu cel alăturat.

Exemplu:

Pentru n=3 avem:

```
1
1 2
1 2 3
```

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int N,i,j;
    cout<<"Introducem N: \t"; cin>>N;
    for(i=1;i<=N;i++){
        for(j=1;j<=i;j++){
            cout<<j<<" ";
        }
        cout<<endl;
    }
}
```

Problema 4.1 - 08

Elaborați un program C/C++ care va permite citirea unui număr natural N și afișează un triunghi de numere asemenea cu cel alăturat.

Exemplu:

Pentru n=2 avem:

```
1
2 3
```

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int N,i,j,k,t=1,sp;
    cout<<"Introducem N: \t"; cin>>N; sp=N+4-1;
    for(i=1;i<=N;i++){
        for(k=sp;k>=1;k--){ cout<<" "; }
        for(j=1;j<=i;j++){
            cout<<t++<<" "; cout<<endl; sp--;
        }
    }
}
```

Problema 4.1 - 09

Elaborați un program C/C++ care va permite afișarea primilor n termeni ai șirului și apoi va afișa soluția expresiei:

$$E=x^1-x^3+x^5-x^7+x^9-x^{11}+x^{13}-x^{15}+....$$

Implementarea C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
    int N,x,sum,m,ctr,i,mm,nn;
    cout<<"Introducem N si x: \t"; cin>>N>>x;
    cout<<"Termenii sunt: \t";
    sum=x; m=-1; cout<<x<<" ";
    for(i=1; i<N; i++){
        ctr=(2*i+1); mm=pow(x,ctr); nn=mm*m; cout<<nn<<" ";
        sum+=nn; m*=(-1);
    }
    cout<<"\nSuma:"<<sum;
}
```

4.2 Aplicarea instrucțiunilor WHILE și DO ... WHILE



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit instrucțiunile ciclice WHILE și DO ... WHILE.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 4.2 - 01

Elaborați un program C/C++ care va permite determinarea numărului de cifre ale unui număr.

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre>#include <iostream> using namespace std; int main(){ int N,cifre=0,k; cout<<"Introducem N: "; cin>>N; k=N; while (N>9){ cifre+= 1;N/=10; } cout<<"Numarul "<<k<<" are "<<cifre+1<<" cifre."; }</pre>	<pre>#include <iostream> using namespace std; int main(){ int N,cifre=0,k; cout<<"Introducem N: "; cin>>N; k=N; do { cifre+= 1;N/=10; } while (N>9); cout<<"Numarul "<<k<<" are "<<cifre+1<<" cifre."; }</pre>

Problema 4.2 - 02

Elaborați un program C/C++ care va permite determinarea sumei numerelor pozitive.

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre>#include <iostream></pre>	<pre>#include <iostream></pre>

<pre>using namespace std; int main() { int n=0,s=0; while (n>=0){ cout<<"Introducem N: "; cin>>n; s+=n; } cout<<"\nSuma: "<<s<<endl; return 0; }</pre>	<pre>using namespace std; int main() { int n=0,s=0; do { cout<<"Introducem N: "; cin>>n; s+=n; } while (n>=0); cout<<"\nSuma: "<<s<<endl; }</pre>
---	--

Problema 4.2 - 03

Elaborați un program C/C++ care va permite determinarea celui mai mare divizor comun a două numere naturale conform algoritmului lui Euclid.

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre>#include <iostream> using namespace std; int main(){ int a,b; cout<<"Introducem a si b: "; cin>>a>>b; int a1=a,b1=b; while (b!=0){ int r=a%b; a=b; b=r; } cout<<"CMMDC("<<a1<<","<<b1; cout<<") = "<<a<<endl; }</pre>	<pre>#include <iostream> using namespace std; int main(){ int a,b; cout<<"Introducem a si b: "; cin>>a>>b; int a1=a,b1=b; do{ int r=a%b; a=b; b=r; } while (b!=0); cout<<"CMMDC("<<a1<<","<<b1; cout<<") = "<<a<<endl; }</pre>

Problema 4.2 - 04

Elaborați un program C/C++ care va permite calcularea sumei și produsului primelor n numere naturale.

Implementarea C++ (while)	Implementarea C++ (do ... while)
#include <iostream>	#include <iostream>

<pre>using namespace std; int main(){ int N,i=1,s=0,p=1; cout<<"Introducem N: "; cin>>N; while(i<=N){ s+=i; p*=i; i++; } cout<<"Suma primelor "<<N; cout<<" numere naturale: \t"; cout<<s<<endl; cout<<"Produsul primelor "<<N; cout<<" numere naturale: \t"; cout<<p<<endl; }</pre>	<pre>using namespace std; int main(){ int N,i=1,s=0,p=1; cout<<"Introducem N: "; cin>>N; do{ s+=i; p*=i; i++; } while (i<=N); cout<<"Suma primelor "<<N; cout<<" numere naturale: \t"; cout<<s<<endl; cout<<"Produsul primelor "<<N; cout<<" numere naturale: \t"; cout<<p<<endl; }</pre>
---	--

Problema 4.2 - 05

Elaborați un program C/C++ care va permite afișarea inversului unui număr întreg pozitiv introdus de la tastatură.

Exemplu:

- Dacă $N = 24\,689$, atunci inversul său va fi: $98\,642$.

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre>#include <iostream> using namespace std; int main(){ int N,inv=0; cout<<"Introducem N: "; cin>>N; // Un alt mod de a scrie o in- structiune while while(inv*=10,inv+=N%10, N/=10); cout<<"Inversul este: \t"; cout<<inv<<endl; }</pre>	<pre>#include <iostream> using namespace std; int main(){ int N,inv=0; cout<<"Introducem N: "; cin>>N; // Un alt mod de a scrie o in- structiune do ... while do(inv*=10,inv+=N%10, N/=10); while(N!=0); cout<<"Inversul este: \t"; cout<<inv<<endl; }</pre>

Problema 4.2 - 06

Elaborați un program C/C++ care va permite afișarea sumei și produsului cifrelor unui număr natural N dat.

Exemplu:

- Dacă $N = 123\,456\,789$, atunci $s = 45$ și $p = 362\,880$.

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre>#include <iostream> using namespace std; int main(){ long long N,s=0,p=1; cout<<"Introducem N: ";cin>>N; while(N!=0){ s+=N%10; p*=N%10; N/=10; } cout<<"\nSuma: \t\t"<<s; cout<<"\nProdusul: \t"<<p; }</pre>	<pre>#include <iostream> using namespace std; int main(){ long long N,s=0,p=1; cout<<"Introducem N: ";cin>>N; do{ s+=N%10; p*=N%10; N/=10; } while(N!=0); cout<<"\nSuma: \t\t"<<s; cout<<"\nProdusul: \t"<<p; }</pre>

Problema 4.2 - 07

Elaborați un program C/C++ care va permite transformarea unui număr N din baza 10 într-o bază B , unde $2 \leq B < 10$, iar rezultatul va fi scris ca inversat.

Exemplu:

- Dacă $N = 1234_{10}$ și $B = 8$, se va afișa 2232_8 (real: 2322_8).

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre>#include <iostream> using namespace std; int main(){ int N,B,y=0; cout<<"Introducem N: \t"; cin>>N; int k=N; cout<<"Introducem B: \t"; cin>>B; while(N!=0){</pre>	<pre>#include <iostream> using namespace std; int main(){ int N,B,y=0; cout<<"Introducem N: \t"; cin>>N; int k=N; cout<<"Introducem B: \t"; cin>>B; do{</pre>

<pre> y=y*10+N%B; N/=B; } cout<<"\nNumarul " <<k; cout<<" in baza " <<B<<" : " <<y; } </pre>	<pre> y=y*10+N%B; N/=B; } while(N!=0); cout<<"\nNumarul " <<k; cout<<" in baza " <<B<<" : " <<y; } </pre>
--	---

Problema 4.2 - 08

Elaborați un program C/C++ care va permite citirea unui text format din cuvinte ce pot fi despartite prin spațiu sau virgulă. Textul citit se termină cu punct sau enter. Să se contorizeze vocalele și consoanele din acel text.

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre> #include<iostream> using namespace std; int main(){ int v=0,c=0; cout<<"Introduceti textul: "; char k; k=getchar(); while(k!='.' && k!='\n'){ if (k!=", ' && k!=' '){ switch (k){ case 'a': case 'A': case 'e': case 'E': case 'i': case 'I': case 'o': case 'O': case 'u': case 'U': v++; break; default: c++; } k=getchar(); } //următorul caracter din text va //fi trimis în variabila k } cout<<"\nNumarul de vocale: " <<v; cout<<"\nNumarul de consoane: " <<c; } </pre>	<pre> #include<iostream> using namespace std; int main(){ int v=0,c=0; cout<<"Introduceti textul: "; char k; k=getchar(); do{ if (k!=", ' && k!=' '){ switch (k){ case 'a': case 'A': case 'e': case 'E': case 'i': case 'I': case 'o': case 'O': case 'u': case 'U': v++; break; default: c++; } k=getchar(); } //următorul caracter din text va //fi trimis în variabila k } while(k!='.' && k!='\n'); cout<<"\nNumarul de vocale: " <<v; cout<<"\nNumarul de consoane: " <<c; } </pre>

Problema 4.2 - 09

Elaborați un program C/C++ care va permite găsirea tuturor numelor perfecte din intervalul [1, 500].

Exemplu:

- Numărul $N=6$ se numește perfect, deoarece este egal cu suma divizorilor săi mai mici ca N : $1+2+3=6$.

Implementarea C++ (while)	Implementarea C++ (do ... while)
<pre>#include <iostream> using namespace std; int main(){ int i=1,u=1,sum=0; cout<<"Numere perfecte sunt: "; cout<<endl; while (i<=500){ while (u<=500){ if (u<i){ if (i%u==0) sum+=u; } u++; } if (sum==i){ cout<<i<<" "; } i++; u=1; sum=0; } }</pre>	<pre>#include <iostream> using namespace std; int main(){ int i=1,u=1,sum=0; cout<<"Numere perfecte sunt: "; cout<<endl; do{ do{ if (u<i){ if (i%u==0) sum+=u; } u++; } while (u<=500); if (sum==i){ cout<<i<<" "; } i++; u=1; sum=0; } while (i<=500); }</pre>

4.3 Particularități ale instrucțiunilor ciclice



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit instrucțiunile `BREAK`, `CONTINUE` și `GOTO`.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 4.3 - 01

Fie următoarea situație: trebuie să scrieți un program care tipărește numerele de la 1 la 10 și nu și cifra 6. Se specifică că trebuie să faceți acest lucru folosind o buclă și anume, o singură buclă este permisă să o utilizați. Explicație: aici vom utiliza instrucțiunea `continue`. Putem rula o buclă de la 1 la 10. De fiecare dată când trebuie să comparăm valoarea iteratorului cu 6, vom folosi instrucțiunea `continue`, pentru a continua la următoarea iterație.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    for (int i = 1; i <= 10; i++) {
        // Dacă i este egal cu 6, continuăți cu următoarea iterație fără a imprima
        if (i==6) continue;
        else cout<<i<<" "; // altfel tipărim valoarea lui i
    }
}
```

Problema 4.3 - 02

Elaborați un program C/C++ care va permite afișarea din intregul alfabet doar literele, caractere mari (mici) până la 'M'. Bucla se va încheia în mod voit, prin folosirea instrucțiunii `"break"`.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    char litera;
    cout<<"Litere majuscule: \t";
    for(litera='A'; litera<='Z'; litera++){
        if (litera=='M') break; cout<<litera<<" ";
    }
    cout<<"\nLitere minuscule: \t";
    for(litera='a'; litera<='z'; litera++){
        if (litera=='m') break; cout<<litera<<" ";
    }
}
```

Problema 4.3 - 03

Elaborați un program C/C++ care va permite calcularea sumei numerelor (maxim 10 numere). Dacă utilizatorul introduce un număr negativ, bucla se termină.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int i; double N, s=0.0;
    for (i = 1; i <= 10; ++i) {
        cout<<"Numarul "<<i<<" este: "; cin>>N;
        // dacă utilizatorul introduce un număr negativ, atunci în-
        trerupem bucla
        if (N < 0.0) break;
        s+=N; // s= s + N;
    }
    cout<<"Suma = "<<s;
}
```

Problema 4.3 - 04

Elaborați un program C/C++ care va permite calcularea sumei numerelor (maxim 10 numere). Dacă utilizatorul introduce un număr negativ, acest număr nu se va adăuga la rezultat.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int i; double N, s=0.0;
    for (i = 1; i <= 10; ++i) {
        cout<<"Numarul "<<i<<" este: "; cin>>N;
        // dacă utilizatorul introduce un număr negativ, atunci nu
        întrerupem bucla, dar nu vom lua în calcul valoarea acestui număr
        if (N < 0.0) continue;
        s+=N; // s= s + N;
    }
    cout<<"Suma = "<<s;
}
```

Problema 4.3 - 05

Elaborați un program C/C++ care va permite introducerea unui număr de la utilizator și verifică dacă numărul dat este prim sau nu.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int N,prim=1,i;
    cout<<"Introducem N: \t";
    cin>>N; int k=N;
    for(i=2; i<N; i++){
        // Dacă numărul este împărțit la orice număr între 2 și n.
        Atunci numărul dat nu este prim.
        if(N % i == 0){
            // Numărul nu este prim. Prin urmare, setăm prim ca 0.
            prim = 0; break;
        }
    }
    if(prim == 1){
        cout<<"\tNumarul este prim.";
    }
    else{
        cout<<"\tNumarul este compus.";
    }
}
```

Problema 4.3 - 06

Elaborați un program C/C++ care va permite utilizatorului să introducă orice valoare întreagă. Apoi va afișa numerele pare și impare în interiorul valorii întregi, din intervalul $[0, n]$.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int N,i;
    cout<<"Introducem N: \t";
    cin>>N;
    for(i=1;i<=N;i++){
        if(i%2!=0){
            cout<<"\nNumarul impar: \t"<<i;
            continue;
        }
        cout<<"\nNumarul par: \t\t"<<i;
    }
}
```

Problema 4.3 - 07

Elaborați un program C/C++ care va permite utilizatorului să introducă două valori întregi. Apoi va afișa valoare maximă.

Implementarea C++

```
#include <iostream>
using namespace std;
int main() {
    int i,j;
    cout<<"Introducem numerele: "; cin>>i>>j;
    if(i>j) goto iMaxim;
    else goto jMaxim;
iMaxim:
    cout<<i<<" este mai mare ca "<<j;
jMaxim:
    cout<<j<<" este mai mare ca "<<i;
}
```

Problema 4.3 - 08

Elaborați un program C/C++ care va permite verificarea dacă un an este bisect sau nu. Fie N , un număr natural ce reprezintă un an.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int an;
    // Etichetăm pentru a rula din nou programul de aici
    INIT:
    cout<<"\nIntroducem anul: "; cin>>an;
    // Folosim operatorul ternar
    ( (an%100!=0 && an%4==0) || (an%400==0) ) ?
    cout<<an<<" este an bisect."<<endl :
    cout<<an<<" nu este an bisect."<<endl;
    // Condiție pentru a merge la eticheta INIT sau
    // Condiție pentru a merge la sfârșitul programului
    if(an!=0) goto INIT; // aceasta este declarația goto
}
```

Problema 4.3 - 09

Elaborați un program C/C++ care va permite calcularea sumei și mediei aritmetice a numerelor pozitive. Dacă se introduce un număr negativ, se va afișa suma și media aritmetică.

Implementarea C++

```
#include <iostream>
#define maxI 100
using namespace std;
int main() {
    int i;
    double n,media,suma = 0.0;
    for (i=1; i<=maxI; ++i){
        cout<<"Introducem N: "; cin>>n;
        // Sari la eticheta soluție,
        // dacă este introdus un număr negativ
        if (n<0.0){
            goto solutie; // aceasta este declarația goto
        }
        suma+=n;
    }
}
```

```

    }
    solutie: // aceasta este o etichetă
        media=suma/(i-1);
    cout<<"\nSuma este: \t"<<suma;
    cout<<"\nMedia este: \t"<<media;
}

```

Problema 4.3 - 10

Elaborați un program C/C++ care va permite calcularea radicalului dintr-un număr.

Implementarea C++

```

#include <iostream>
#include <cmath> // pentru funcția sqrt(), adică radical
using namespace std;
int main(){
    double x;
    MaiIncearca: // aceasta este o etichetă
        cout<<"Introducem un numar nenegativ: "; cin>>x;
    if (x<0.0) goto MaiIncearca; // aceasta este declarația goto
    cout<<"Radical din "<<x<<" este "<<sqrt(x)<<endl;
    return 0;
}

```

Problema 4.3 - 11

Elaborați un program C/C++ care va permite determinarea dacă un număr este par sau impar.

Implementarea C++

```

#include <iostream>
using namespace std;
int main(){
    int x;
    cout<<"Introducem un numar: "; cin>>x;
    if (x%2==0)goto par; // aceasta este declarația goto
    else goto impar; // aceasta este declarația goto
    par: cout << x << " este par"; exit(0);
    impar: cout << x << " este impar"; exit(0);
}

```

4.4 Probleme propuse

1. Probleme de aplicare a instrucțiunii FOR

1. *Elaborați un cod C/C++ care va afișa valoarea termenului al N-lea al seriei, dacă N este un număr natural:*
 - a) $S=1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots, \frac{1}{2^{n-1}};$
 - b) $S=1, 4, 7, 10, \dots, 3 \cdot n - 2;$
 - c) $S=\frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}, \frac{1}{11}, \dots, \frac{1}{2 \cdot n - 1};$
 - d) $S=2, 8, 24, 64, 160, \dots, n \cdot 2^n;$
2. *Elaborați un cod C/C++ care va afișa valoarea următoarei sume, dacă N este un număr natural:*
 - a) $S=1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\dots+\frac{1}{2^{n-1}};$
 - b) $S=1 \cdot 2 + 2 \cdot 3 + \dots + n \cdot (n+1);$
 - c) $S=1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\dots+(-1)^{n+1}\frac{1}{n};$
 - d) $S=1 \cdot 4 + 2 \cdot 9 + \dots + n \cdot (5 \cdot n - 1);$
3. *Elaborați un cod C/C++ care va afișa în limita intervalului $[0, N]$, unde N este un număr întreg pozitiv:*
 - a) toate numerele pare, împreună cu suma și produsul acestora;
 - b) toate numerele impare, împreună cu suma și produsul acestora;
 - c) toate numerele perfecte, împreună cu suma și produsul acestora;
 - d) toate numerele prime, împreună cu suma și produsul acestora.
4. *Elaborați un cod C/C++ care va afișa:*
 - a) media aritmetică a primelor N numere perfecte;
 - b) câte dintre N numere au suma cifrelor un număr prim;
 - c) cea mai mare cifră primă și de câte ori apare printre cele N numere citite de la tastatură, dacă nu există nici un număr prim se va afișa -1.
5. *Elaborați un cod C/C++ care va afișa primele N numere perfecte, dacă $(N \leq 5)$ este un număr natural, implementați algoritmul lui Euclid pentru determinarea numerelor perfecte.*

Explicație: Euclid a constatat: dacă $2^n - 1$ este un număr prim, atunci numărul $2^{n-1} \cdot 2^n - 1$.

6. *Elaborați un cod C/C++ care va afișa toți divizorii unui număr natural N , numărul de divizori ai acestui număr, suma și produsul divizorilor acestuia.*
7. *Elaborați un cod C/C++ care va afișa un mesaj corespunzător dacă un număr natural N este pseudoperfect. Un număr este pseudoperfect dacă este divizor al sumei divizorilor săi.*
8. *Elaborați un cod C/C++ care va afișa un mesaj corespunzător dacă două numere naturale a și b sunt prietene. Două numere se numesc prietene dacă fiecare număr este egal cu suma divizorilor celuilalt număr.*
9. *Elaborați un cod C/C++ care va afișa dacă un număr natural nenul N cu maxim 9 cifre obținut după eliminarea cifrelor sale pare este prim.*
10. *Elaborați un cod C/C++ care va afișa dacă un număr natural nenul N cu maxim 9 cifre obținut după eliminarea cifrelor sale impare este perfect.*
11. *Elaborați un cod C/C++ care va afișa dacă un număr natural nenul N este aproape prim. Un număr este aproape prim dacă se poate scrie ca produs de două numere prime distincte.*
12. *Elaborați un cod C/C++ care va determina primii N termeni ai progresiei aritmetice, dacă se cunosc valorile următorilor termeni: $a_3 = 2$ și $a_5 = -2$.*
13. *Elaborați un cod C/C++ care va determina numerele, ce sunt termeni comuni ai progresiilor aritmetice: 2, 5, 8, ..., 332 și 7, 12, 17, ..., 157.*
14. *Elaborați un cod C/C++ care va afișa divizorii proprii ai numărului natural N , unde $N < 2\,000\,000\,000$ și numărul acestora. Exemplu: pentru $N=30$, divizori proprii sunt: 2, 3, 5, 6, 10, 15, iar numărul lor total este 6.*
15. *Elaborați un cod C/C++ care va afișa în ordine descrescătoare toți divizorii numărului natural N , unde $N > 1$. Exemplu: $N=10$, afișăm: 10, 5, 2, 1.*
16. *Elaborați un cod C/C++ care va afișa toate numerele prime până la N , unde $N > 2$. Exemplu: pentru $n=10$, afișăm numerele 2, 3, 5, 7.*
17. *Elaborați un cod C/C++ care va afișa toate perechile de numere naturale (x,y) pâna la numărul natural N , ($1 < x < y \leq N$), pentru care x este divizorul lui y , numerele x și y au aceeași cifră a unităților. Exemplu: pentru $n=30$, afișăm: (2,12), (2,22), (4,24), (5,15), (5,25), (10,20), (10,30).*
18. *Elaborați un cod C/C++ care va afișa valorile funcției sinus între două limite $[a, b]$ și un pas E , specificate de utilizator.*
19. *Elaborați un cod C/C++ care va afișa a^n , fără a aplica funcțiile din biblioteca math.h sau cmath, unde n și a sunt două numere întregi.*
20. *Elaborați un cod C/C++ care va afișa toate numerele din intervalul $[a, b]$, unde ($a < b$, $a > 1$, $b < 1000$), care au un număr par de divizori proprii. Exemplu: pentru $a=10$ și $b=25$, afișăm: 10, 12, 14, 15, 18, 20, 21, 22, 24.*

2. Probleme de aplicare a instrucțiunii WHILE și DO ... WHILE

1. *Elaborați un cod C/C++ care va citi un număr natural N, apoi se citesc N numere întregi cu cel puțin 9 cifre fiecare pentru a realiza următoarele operațiuni:*
 - a) *afișarea numărului care are cea mai mare (mică) sumă a cifrelor sale;*
 - b) *afișarea numărului care are cel mai mare (mic) produs a cifrelor sale;*
 - c) *determină cea mai mare (mică) cifră ce ale celor N numere citite;*
 - d) *ordonează ascendent (descendent) cele N numere în raport cu suma cifrelor acestor numere, dacă un număr va fi negativ, atunci prima cifră se va considera negativă în calculul sumei cifrelor numărului dat.*
2. *Elaborați un cod C/C++ care va afișa numărul cifrelor egale cu 1 care apar în scrierea fiecărui număr natural N în baza 2, unde $1000 < N < 10000$ reprezintă numărul de numere naturale citite de la tastatură.*
3. *Elaborați un cod C/C++ care va afișa cifrele pare (impare) ale unui număr natural N în ordinea inversă apariției, separate prin caracterul "#".*
4. *Elaborați un cod C/C++ care va forma două noi numere, unul format din cifrele pare ale numărului natural N cu cel puțin 4 cifre, celălalt format din cifrele impare.*
5. *Elaborați un cod C/C++ care va afișa numărul de valori prime (pare, impare) citite, dacă se citesc numere întregi până la întâlnirea valorii 0.*
6. *Elaborați un cod C/C++ care va afișa cele două numere formate prin "înjumătățirea" scrierii zecimale a numărului natural n, unde $n > 10\,000$. Exemplu: Pentru $n=12\,345$ se va afișa 12 și 345, iar pentru $n=182\,345$ se va afișa 182 și 345.*
7. *Elaborați un cod C/C++ care va afișa cifrele situate pe poziții impare începând cu cifra unităților (de la dreapta spre stânga) ale unui număr natural n, unde $n > 1000$. Exemplu: pentru $n=12\,345$ se va afișa 52.*
8. *Elaborați un cod C/C++ care va afișa suma primelor două cifre (ultimelor două cifre) ale unui număr natural n, unde $n > 1000$. Exemplu: pentru $n=1235$ se va afișa 3.*
9. *Elaborați un cod C/C++ care va afișa cel mai mic multiplu par al numărului format din prima și ultima cifră a unui număr natural n ce conține minim 3 cifre. Exemplu: pentru $n=1235$ se va afișa 30*

10. *Elaborați un cod C/C++ care va afișa numărul format din cifrele pare (impare) ale unui număr natural n , unde $n > 1000$, situate pe poziții impare (pare) începând cu prima (ultima) cifră a sa. Exemplu: pentru $n = 724582$ se va afișa 48.*
11. *Elaborați un cod C/C++ care va afișa numărul de apariții ale cifrei unităților în scrierea numărului natural n , unde $n > 10\,000$. Exemplu: pentru $n = 15\,535$ se va afișa 3 (5 apare de 3 ori)*
12. *Elaborați un cod C/C++ care va afișa cea mai mare cifră care apare în scrierea unui număr natural n și numărul de apariții al ei. Exemplu: pentru $n = 19\,539$ se va afișa "9 apare de 2 ori".*
13. *Elaborați un cod C/C++ care va afișa numărul n după înlocuirea tuturor cifrelor $c1$ cu cifra $c2$, dacă se știe că la început se citește un număr natural n cu cel mult 9 cifre, apoi se introduc de la tastatură două valori $c1$ și $c2$ reprezentând două cifre, se știe că $c2$ este totdeauna o cifră nenulă!*
14. *Elaborați un cod C/C++ care va afișa cea mai din dreapta cifră pară și cea mai din stânga cifră impară ale unui număr natural n ce conține maxim 9 cifre. Dacă n nu conține cifre pare se va afișa -1, iar dacă nu conține cifre impare se va afișa -2. Exemplu: pentru $n = 1278$ se va afișa: 8 1.*
15. *Elaborați un cod C/C++ care va afișa numărul după efectuarea următoarelor schimbări: se inversează prima cifră cu a doua, a treia cu a patra, etc. Dacă numărul n este format dintr-un număr impar de cifre se va afișa mesajul nu se poate efectua. Exemplu: $n = 1234$ se va afișa: 2143.*
16. *Elaborați un cod C/C++ care va afișa mesajul „Este palindrom” sau „Nu este palindrom”, dacă numărul natural n ce conține maxim 10 cifre, are sau nu această proprietate. Un număr este palindrom dacă este egal cu oglinditul său. Exemplu: pentru $n = 1\,234\,321$ se va afișa este palindrom.*
17. ****Cifra de control se poate obține astfel: se efectuează suma cifrelor numărului natural n dat. Dacă aceasta are mai mult de o cifră, n devine suma cifrelor sale și se continuă algoritmul până când n ajunge de o cifră. Aceasta este cifra de control. Exemplu: pentru $n = 247$ se obțin pe rând valorile: $13 (2+4+7)$, $4 (1+3)$. Se va afișa 4.*
 - a) *Elaborați un cod C/C++ care va determina cifra de control folosind o singură structură repetitivă;*
 - b) *Elaborați un cod C/C++ care va determina cifra de control fără nicio structură repetitivă utilizată.*

3. Probleme de aplicare a particularităților instrucțiunilor ciclice



- ✓ Declarația *break* are două forme: etichetată și neetichetată. O instrucțiune *break* fără etichetă încheie comutatorul *switch*, *for*, *while* sau *do-while*, dar un *break* etichetat termină o instrucțiune externă.
- ✓ Instrucțiunea *continue* omite iterația curentă a unei bucle *for*, *while* sau *do-while*. Forma fără etichetă sare până la capătul corpului buclei celei mai interioare și evaluează expresia booleană care controlează bucla. O declarație *continue* etichetată omite iterația curentă a unei bucle exterioare marcate cu eticheta dată.
- ✓ Aceste instrucțiuni se pot aplica problemelor deja propuse la temele precedente, așa cum am menționat și pentru instrucțiunea *IF* sau *SWITCH*.

1. Elaborați un program în limbajul C++ care va calcula valoarea funcției:

$$a) \quad y = \begin{cases} x, & \text{dacă } x \geq 3; \\ 2 \cdot x, & \text{dacă } x < 3; \end{cases}$$

$$e) \quad y = \begin{cases} x+1, & \text{dacă } x > 7; \\ 5 \cdot x - 1, & \text{dacă } x \leq 7; \end{cases}$$

$$b) \quad y = \begin{cases} x^3 + 1, & \text{dacă } x \geq 0; \\ \ln(e^{x+1}), & \text{dacă } x < 0; \end{cases}$$

$$f) \quad y = \begin{cases} \lg(10 \cdot x), & \text{dacă } x \geq 2; \\ \sqrt{|x|}, & \text{dacă } x < 2; \end{cases}$$

$$c) \quad y = \begin{cases} x^2 + 5, & \text{dacă } x \geq 5; \\ \lg(e^{x+1} + 1), & \text{dacă } x < 5; \end{cases}$$

$$g) \quad y = \begin{cases} \ln\left(\frac{10 \cdot x}{3}\right), & \text{dacă } x \geq 4; \\ \sqrt{|x-3|} + x^3, & \text{dacă } x < 4; \end{cases}$$

$$d) \quad y = \begin{cases} x^3 + \frac{2}{5}, & \text{dacă } x \geq 1; \\ \ln(e^{x-2} + x), & \text{dacă } x < 1; \end{cases}$$

$$h) \quad y = \begin{cases} \frac{x}{7} + \ln(x+1), & \text{dacă } x \geq 6; \\ \frac{2 \cdot x^3}{5}, & \text{dacă } x < 6; \end{cases}$$

2. Elaborați un program în limbajul C++ care va afișa mesajul „Foarte bine” dacă un număr real N este mai mare sau egală cu 8, mesajul „Bine” dacă N este mai mic decât 8, dar mai mare sau egal ca 5 și mesajul „Insuficient”, dacă N este mai mic decât 5.

3. *Elaborați un program în limbajul C++ care va afișa mesajul „Promovat”, dacă nota este mai mare sau egală cu 5 și „Ne promovat” - în caz contrar. Se citește un număr real nenul și pozitiv n reprezentând nota medie la Informatică a unui elev. Stabiliți dacă elevul a fost promovat.*
4. *Elaborați un program în limbajul C++ care va citi de la tastatură un cuvânt format din minim 9 caractere, apoi realizați următoarele operațiuni:*
 - a) *afișați câte vocale și câte consoane conține cuvântul;*
 - b) *afișați de câte ori se repetă o anumită vocală O (sau consoană C).*
5. *Elaborați un program în limbajul C++ care va citi de la tastatură un număr natural a cu cel puțin 8 cifre, apoi realizați următoarele operațiuni:*
 - a) *afișați câte cifre pare și câte cifre impare are acest număr;*
 - b) *afișați de câte ori se repetă o cifră pară P (sau impară I).*
6. *Elaborați un program în limbajul C++ care va citi de la tastatură două numere naturale nenule A și B cu câte 3 cifre fiecare, apoi realizați următoarele operațiuni:*
 - a) *afișați câte dintre cifrele numărului A sunt strict mai mari (sau mai mici) decât toate cifrele lui B și afișați suma acestor cifre;*
 - b) *afișați câte dintre cifrele numărului B sunt strict mai mari (sau mai mici) decât toate cifrele lui A și afișați produsul acestor cifre.*
7. *Elaborați un program în limbajul C++ care va afișa media aritmetică a două numere naturale nenule a și b și intervalul în care se află. Pe primul rând al ecranului se va afișa media aritmetică cu două zecimale, iar pe al doilea rând - extremitățile intervalului în care se situează media calculată. Intervalele disponibile sunt: [0,4), [4,7), [7,10].*
8. *Elaborați un program în limbajul C++ care va afișa ultima cifră a expresiei $S+P$, unde S este suma primelor 2 numere, iar P este produsul ultimelor două. Cele 3 numere întregi pozitive nenule sunt citite de la tastatură.*
9. *Elaborați un program în limbajul C++ care va verifica dacă 3 numere reale pozitive nenule pot reprezenta laturile unui triunghi, caz în care afișați pe câte un rând al ecranului, aria și perimetrul triunghiului. În cazul în care cele 3 numere nu pot fi laturile unui triunghi, afișați un mesaj corespunzător.*
10. *Elaborați un program în limbajul C++ care va verifica dacă 3 numere întregi pozitive nenule pot reprezenta coeficienții ecuației lui Pitagora: $X^2+Y^2=Z^2$. În cazul afirmativ, afișați mesajul: „ Sunt numere pitagorice”.*

REZOLVAREA PROBLEMELOR CU ARRAY-URI ÎN LIMBAJUL C/C++

5.1 Aplicarea array-urilor unidimensionale



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte. Un array unidimensional poate fi denumit vector, iar un array bidimensional – matrice.
- ✓ În acest compartiment vom aplica în mod implicit array-urile unidimensionale (vectorii) la rezolvarea problemelor de diverse nivele de complexitate.
- ✓ Vom pune accentul pe următoarele operațiuni efectuate asupra vectorilor: parcurgerea cu prelucrarea componentelor; căutarea secvențială; inserarea; ștergerea și ordonarea componentelor.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 5.1 - 01

Elaborați un cod C/C++ care va permite introducerea de la tastatură a componentelor unui vector și le va afișa la ecran în ordinea introducerii acestora.

Implementarea C++

```
#include <iostream>
using namespace std;
int main() {
    // n - este dimensiunea vectorului;
    // i - este reprezentarea indicilor componentelor;
    int v[100],n,i;
    cout<<"Introducem dimensiunea vectorului, n= ";cin>>n;
    cout<<"Introducem componentele vectorului: "<<endl;
    for(i=0;i<n;i++){
```

```

        cout<<"\tv["<<i+1<<"]= "; cin>>v[i];
    }
    cout<<"Afișam componentele vectorului: \n\t";
    for(i=0;i<n;i++){
        cout<<v[i]<<" ";
    }
}

```

Problema 5.1 - 02

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod și va afișa la ecran media aritmetică a componentelor.

Implementarea C++

```

#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    int v[]={1,5,20,6,9,12,15,3,10}; // componentele vectorului
    int n=sizeof v/sizeof v[0]; // determinam numarul componentelor
    int i,s=0; // declaram indicele si suma initiala
    cout<<"Parcurgem fiecare componenta si determinam suma!"<<endl;
    for(i=0;i<n;i++){
        s+=v[i];
    }
    cout<<"Afișam suma si media componentele vectorului: \n\t";
    /* fixed<<setprecision(3) - ne permite afișarea cu 3 zecimale
    după virgula; operatia "/" ne permite determinarea partii întregi,
    dar în cazul respectiv noi avem nevoie si de partea fractionara, de
    aceea vom converti cu (float) numarul s/n. */
    cout<<s<<" "<<fixed<<setprecision(3)<<(float) s/n;
}

```

Problema 5.1 - 03

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod și va afișa la ecran componenta maximă și cea minimă.

Implementarea C++

```

#include <iostream>
using namespace std;
int main() {

```

```

int v[]={1,-5,20,-6,9,12,15,-3,10},n=sizeof v/sizeof v[0],i;
int minim=v[0],maxim=v[0]; // declaram minim si maxim
cout<<"Parcurgem fiecare componenta: "<<endl;
for(i=0;i<n;i++){
    if(v[i]<minim) minim=v[i];
    else if (v[i]>maxim) maxim=v[i];
}
cout<<"\tElementul minim din vector: "<<minim<<endl;
cout<<"\tElementul maxim din vector: "<<maxim<<endl;
}

```

Problema 5.1 - 04

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod și va afișa la ecran un mesaj care va confirma dacă componentele sunt aranjate în ordine ascendentă.

Implementarea C++

```

#include <iostream>
using namespace std;
int main(){
    int v[]={10,49,35,25,6,3},n=sizeof v/sizeof v[0],i=0;
    int ascendent=1;//presupunem ca vectorul este ordonat crescator
    while((i<n-1)&&(ascendent!=0)){
        if(v[i]>v[i+1]) ascendent=0;
        i++;
    }
    if(ascendent==0) cout<<"Vectorul nu este ordonat ascendent!";
    else cout<<"Vectorul este ordonat crescator!";
}

```

Problema 5.1 - 05

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod și va afișa la ecran componentele pare și cele impare.

Implementarea C++

```

#include <iostream>
using namespace std;
int main() {
    int v[]={1,-5,20,-6,9,12,13,10},n=sizeof v/sizeof v[0],i;
    cout<<"Parcurgem fiecare componenta: "<<endl;
}

```

```

    cout<<"\tComponentele pare sunt: \t";
    for(i=0;i<n;i++){
        if(v[i]%2==0) cout<<v[i]<<"\t";
    }
    cout<<"\n\tComponentele impare sunt: \t";
    for(i=0;i<n;i++){
        if(v[i]%2!=0) cout<<v[i]<<"\t";
    }
}

```

Problema 5.1 - 06

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod și va afișa la ecran valoarea componentei cu număr maxim de divizori. În cazul în care sunt mai multe astfel de componente se va afișa cea cu valoarea cea mai mică (cea mai mare).

Implementarea C++

```

#include <iostream>
using namespace std;
int main() {
    int v[]={10,49,35,25,6,3},n=sizeof v/sizeof v[0],i,k;
    int min_div=0,elem=v[0],nr_div;
    cout<<"Parcurgem fiecare componenta: "<<endl;
    for(i=1;i<=elem;i++){
        if(elem%i==0) min_div++;
    }
    for(i=1;i<n;i++){
        nr_div=0; // numărul de divizori
        for(k=1;k<=v[i];k++){
            if(v[i]%k==0) nr_div++;
            if((nr_div>min_div)&&(elem>v[i])){
                min_div=nr_div; elem=v[i];
            }
        }
    }
    cout<<"\tValoarea obtinuta este: "<<elem;
    cout<<"\n\tNumarul "<<elem<<" are "<<min_div<<" divizori!";
}

```


Problema 5.1 - 07

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod și va afișa la ecran valorile prime, apoi va afișa numerele prime a căror invers este tot un număr prim.

Implementarea C++

```
#include <iostream>
using namespace std;
int v[]={3,17,25,60,13,47},n=sizeof v/sizeof v[0],i=0;
int nr_prime=0,nr_div;
int main(){
    cout<<"Afisarea valorilor prime:";
    for(i=0;i<n;i++){
        nr_div=0;
        for(int j=1;j<=v[i];j++){
            if(v[i]%j==0)nr_div++;
        }
        if(nr_div==2){
            nr_prime++; cout<<v[i]<<" ";
        }
    }
    if(nr_prime==0) cout<<"Nu exista numere prime!"<<endl;
    cout<<"Afisam valorile prime al caror invers este prim!\n";
    for(i=0;i<n;i++) {
        nr_div=0;
        for(int j=1;j<=v[i];j++){
            if(v[i]%j==0)nr_div++;
        }
        if(nr_div==2){
            int x1=v[i],x2=0; // x2 consideram a fi inversul
            //calculam inversul lui v[i] care este x2
            while(x1!=0){
                x2=x2*10+x1%10; x1=x1/10;
            }
            int nr_div1=0; //verificam daca x2 este si el prim
            for(int j=1;j<=x2;j++){
                if(x2%j==0)nr_div1++;
            }
            if(nr_div1==2)cout<<"\t"<<v[i]<<" \t"si"\t "x2<<endl;
        }
    }
}
```

Problema 5.1 – 08

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod. Să se verifice dacă un număr definit aparține vectorului.

Implementarea C++

```
#include <iostream>
using namespace std;
// definim implicit un numar ce va trebui cautat
#define k 4
int main(){
    int v[]={3,17,25,5,12,21,61,13,4};
    //stabilim formula de determinare a dimensiunii vectorului
    int n=sizeof v /sizeof v[0];
    cout<<"Elementele vectorului declarat implicit sunt: \n";
    //afisam elementele vectorului declarat implicit
    for (int i=0;i<n;i++){
        cout<<v[i]<<" ";
    }
    cout<<"\n\nVerificam apartenenta numarului k= "<<k;
    // parcurgem fiecare element al vectorului de date
    for (int i=0;i<n;i++){
        if (i==k) cout<<"\nNumarul "<<k<<" apartine vectorului!\n";
    }
}
```

Problema 5.1 - 09

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod. Să se afișeze indicii componentelor egale cu cea de a doua componentă din vector.

Implementarea C++

```
#include <iostream>
using namespace std;
int main(){
    int v[]={1, -2, 3, -2, 4, -2, 9, 8};
    //stabilim formula de determinare a dimensiunii vectorului
    int i,n=sizeof v /sizeof v[0];
    cout<<"Elementele vectorului declarat implicit sunt: \n\t";
    //afisam elementele vectorului declarat implicit
    for (int i=0;i<n;i++){
        cout<<v[i]<<" ";
    }
```

```

    }
    cout<<"\nElementul "<<v[1]<<" se mai repeta la indicii: \n\t";
    for (i=0;i<n;i++){
        if (v[i]==v[1]) cout<<i<<" ";
    }
}

```

Problema 5.1 - 10

Elaborați un cod C/C++ în care componentele unui vector vor fi declarate implicit în cod. Să se elimine valoarea de pe poziția p din vector.

Implementarea C++

```

#include <iostream>
#define p 6
using namespace std;
int main(){
    int v[]={1, -2, 3, -2, 4, -2, 9, 8};
    int i,n=sizeof v /sizeof v[0];
    cout<<"Elementele vectorului declarat implicit sunt: \n\t";
    //afisam elementele vectorului declarat implicit
    for (i=0;i<n;i++){
        cout<<v[i]<<" ";
    }
    cout<<"\nStergerea valorii de pe pozitia "<<p<<" , adica elementul "<<v[p];
    if (p<0 || p>n) cout<<"Pozitia nu exista in vector!"<<endl;
    else{
        // stergem elementul de pe poziția p
        for (i=p;i<n-1;i++){
            v[i]=v[i+1];
        }
        n--;
        //afisam elementele vectorului după stergere
        cout<<"\nElementele vectorului dupa stergere sunt:\n\t";
        for (i=0;i<n;i++){
            cout<<v[i]<<" ";
        }
    }
}

```

5.2 Aplicarea array-urilor bidimensionale



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit array-urile bidimensionale (matricile) la rezolvarea problemelor de diverse nivele de complexitate.
- ✓ Vom pune accentul pe următoarele operațiuni efectuate asupra matricilor: parcurgerea cu prelucrarea componentelor; căutarea secvențială; inserarea; ștergerea și ordonarea componentelor.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 5.2 - 01

Elaborați un cod C/C++ care va permite introducerea de la tastatură a componentelor unei matrici și le va afișa la ecran în ordinea introducerii acestora, precum și transpusa matricii.

Implementarea C++

```
#include <iostream>
using namespace std;
int main() {
    // n, m - sunt dimensiunile matricii;
    // i, j - reprezinta indicii componentelor;
    int M[20][20], n, m, i, j;
    cout<<"Introducem dimensiunile matricii: "; cin>>n>>m;
    cout<<"Introducem componentele matricii: "<<endl;
    for(i=0; i<n; i++){
        for(j=0; j<m; j++){
            cout<<"\tM["<<i+1<<"]["<<j+1<<"]=" "; cin>>M[i][j];
        }
    }
    cout<<"Afișam componentele matricii: \n";
    for(i=0; i<n; i++){
        for(j=0; j<m; j++){
            cout<<" "<<M[i][j];
        }
    }
    cout<<endl;
```

```

    }
    cout<<"Afisam transpusa matricii: \n";
    for(j=0;j<m;j++){
        for(i=0;i<n;i++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
}

```

Problema 5.2 - 02

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod. Afișați matricea obținută după ștergerea unei linii și a unei coloane.

Implementarea C++

```

#include <iostream>
#include <cmath>
#define n 4 // linii
#define m 3 // coloane
using namespace std;
int i,j,nr=0;
int x,y;
int main() {
    // declaram matricea M cu componentele sale
    int M[n][m]={1,2,3},{4,5,6},{7,8,9},{5,7,9}};
    cout<<"Afisam componentele matricii M: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    cout<<endl<<"Indicele liniei de sters incepe de la 0: ";
    cin>>x;
    for(i=x;i<n-1;i++){
        for(j=0;j<m;j++){
            M[i][j]=M[i+1][j];
        }
    }
    for(i=0;i<n-1;i++){
        for(j=0;j<m;j++){

```

```

        cout<<" "<<M[i][j];
    }
    cout<<endl;
}
cout<<endl<<"Indicele coloanei de sters incepe de la 0: ";
cin>>y;
for(i=0;i<n;i++){
    for(j=y;j<m-1;j++){
        M[i][j]=M[i][j+1];
    }
}
for(i=0;i<n-1;i++){
    for(j=0;j<m-1;j++){
        cout<<" "<<M[i][j];
    }
    cout<<endl;
}
}
}

```

Problema 5.2 - 03

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod și va afișa la ecran media aritmetică a componentelor acesteia.

Implementarea C++

```

#include <iostream>
#include <iomanip>
#define n 3 // linii
#define m 4 // coloane
using namespace std;
int main() {
    // declarăm matricea cu componentele sale
    int M[n][m]={1,2,3,4},{5,6,7,8},{9,0,1,2}},i,j,s=0;
    cout<<"Afișăm componentele matricii: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j]; s+=M[i][j]; // calculăm suma
        }
        cout<<endl;
    } // vom utiliza setprecision() din biblioteca iomanip
    cout<<"Afișăm suma și media componentelor matricii: \n\t";
    cout<<s<<" "<<fixed<<setprecision(3)<<(float) s/(n*m);
}

```

```
}
```

Problema 5.2 - 04

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod și va afișa la ecran componenta maximă și cea minimă a matricii.

Implementarea C++

```
#include <iostream>
#define n 3 // linii
#define m 4 // coloane
using namespace std;
int main() {
    // declaram matricea cu componentele sale
    int M[n][m]={1,2,3,4},{5,6,7,8},{9,0,1,2}},i,j;
    //initializam componenta minima si maxima
    int m1=M[0][0],m2=M[0][0];
    cout<<"Parcurgem fiecare componenta: "<<endl;
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if(M[i][j]<m1) m1=M[i][j];
            else if (M[i][j]>m2) m2=M[i][j];
        }
    }
    cout<<"\tElementul minim din matrice: "<<m1<<endl;
    cout<<"\tElementul maxim din matrice: "<<m2<<endl;
}
```

Problema 5.2 - 05

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod și va afișa la ecran componentele pare (sau impare) de pe fiecare linie a matricii.

Implementarea C++

```
#include <iostream>
#define n 3 // linii
#define m 4 // coloane
using namespace std;
int main() {
    int M[n][m]={1,3,5,4},{7,6,8,0},{9,0,1,2}},i,j;
```

```

    cout<<"Parcurgem matricea si afisam componentele pare: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if(M[i][j]%2==0) cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
}

```

Problema 5.2 - 06

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod și va afișa la ecran suma elementelor de pe marginea matricii.

Implementarea C++

```

#include <iostream>
#define n 4 // linii
#define m 4 // coloane
using namespace std;
int main() {
    int M[n][m]={1,2,3,4},{3,7,8,1},{5,6,7,8},{9,1,2,3}},i,j;
    int suma=0;
    cout<<"Afisam componentele matricii: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    cout<<"Suma componentelor de pe marginea matricii: ";
    for(j=0;j<m;j++){
        suma+=M[0][j]; suma+=M[n-1][j]; //linia 1 si linia n
    }
    for(i=1;i<n-1;i++){
        suma+=M[i][0]; suma+=M[i][n-1]; //coloana 1 si coloana m
    }
    cout<<suma<<endl;
}

```


Problema 5.2 - 07

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod și va afișa la ecran maximul fiecărei coloane și minimul fiecărei linii.

Implementarea C++

```
#include <iostream>
#define n 4 // linii
#define m 4 // coloane
using namespace std;
int main() {
    int M[n][m]={1,2,3,4},{3,7,8,1},{5,6,7,8},{9,1,2,3}},i,j;
    int m1,m2;
    cout<<"Afișam componentele matricii: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    cout<<"Componentele minime de pe fiecare linie sunt:\n";
    for(i=0;i<n;i++){
        m1=M[i][0];
        for(j=1;j<m;j++){
            if(M[i][j]<m1) m1=M[i][j];
        }
        cout<<" "<<m1;
    }
    cout<<"\nComponentele maxime de pe fiecare coloana sunt:\n";
    for(j=0;j<m;j++){
        m2=M[0][j];
        for(i=1;i<n;i++){
            if(M[i][j]>m2) m2=M[i][j];
        }
        cout<<" "<<m2;
    }
}
```

Problema 5.2 - 08

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod. Verificați dacă matricea este triunghiulară superior (dacă sub diagonala principală sunt doar zerouri).

Implementarea C++

```
#include <iostream>
#include <cmath>
#define n 3 // linii
#define m 3 // coloane
using namespace std;
int i,j,nr=0;
int main() {
    // declaram matricea M cu componentele sale
    int M[n][m]={1,2,3},{0,6,7},{0,0,1}};
    cout<<"Afisam componentele matricii M: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    // Verificam daca matricea este triunghiulara superior
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            //calculam numarul de elemente nule situate sub diagonala principala
            if (i>j && M[i][j]==0) nr++;
        }
    }
    if (nr==(pow(n,2)-n)/2) cout<<"Este superior triunghiulara.";
    else cout<<"Nu este superior triunghiulara.";
}
```

Problema 5.2 - 09

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod. Verificați dacă matricea este triunghiulară superior (dacă deasupra diagonalei principale sunt doar zerouri).

Implementarea C++

```
#include <iostream>
#include <cmath>
#define n 3 // linii
#define m 3 // coloane
using namespace std;
```

```

int i,j,nr=0;
int main() {
    // declaram matricea M cu componentele sale
    int M[n][m]={1,0,0},{2,6,0},{3,2,1}};
    cout<<"Afisam componentele matricii M: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" " <<M[i][j];
        }
        cout<<endl;
    }
    // Verificam daca matricea este triunghiulara inferior
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            //calculam numarul de elemente nule situate deasupra diagonalei
            //principale
            if (i<j && M[i][j]==0) nr++;
        }
    }
    if (nr==(pow(n,2)-n)/2) cout<<"Este inferior triunghiulara.";
    else cout<<"Nu este inferior triunghiulara.";
}

```

Problema 5.2 - 10

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod și va afișa la ecran componentele de pe diagonala principală și de pe diagonala secundară a matricii.

Implementarea C++

```

#include <iostream>
#define n 3 // linii
#define m 3 // coloane
using namespace std;
int main() {
    int M[n][m]={1,3,5},{7,6,8},{9,1,2}},i,j;
    cout<<"Afisam componentele matricii: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" " <<M[i][j];
        }
        cout<<endl;
    }
}

```

```

    cout<<"\nPrincipala\tSecundara"<<endl;
    for(i=0;i<n;i++){
        cout<<M[i][i]<<"\t\t"<<M[i][n-i-1]<<endl;
    }
}

```

Problema 5.2 - 11

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod. Afișați elementele situate deasupra și sub diagonala principală a matricii.

Implementarea C++

```

#include <iostream>
#include <iomanip>
#define n 4 // linii
#define m 4 // coloane
using namespace std;
int main() {
    // declaram matricea cu componentele sale
    int M[n][m]={1,2,3,4},{3,7,8,1},{5,6,7,8},{9,1,2,3}},i,j;
    cout<<"Afisam componentele matricii: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    cout<<"Componentele deasupra diagonalei principale:\n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if (i<j) cout<<" "<<M[i][j];
        } //cout<<endl; afisarea datelor de pe fiecare rand
    }
    cout<<"\nComponentele sub diagonala principala: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if (i>j) cout<<" "<<M[i][j];
        } //cout<<endl; afisarea datelor de pe fiecare rand
    }
}

```

Problema 5.2 - 12

Elaborați un cod C/C++ în care componentele unei matrici vor fi declarate implicit în cod. Afișați elementele situate deasupra și sub diagonala secundară a matricii. Utilizați componente de colorare text.

Implementarea C++

```
#include <iostream>
#include <iomanip>
#include <dos.h> // selectam functia system("Color XY");
#define n 4 // linii
#define m 4 // coloane
using namespace std;
int main() {
    // declaram matricea cu componentele sale
    int M[n][m]={1,2,3,4},{5,6,7,8},{9,0,1,2},{3,4,5,6},i,j;
    cout<<"Afisam componentele matricii: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    //se va colora textul in aqua cu fundal negru
    getch(); system("CLS"); system("Color 03");
    cout<<"Componentele deasupra diagonalei secundare:\n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if (i+j<n-1){
                cout<<" "<<M[i][j];
            }
        } //cout<<endl; afisarea datelelor de pe fiecare rand
    }
    //se va colora textul in galben cu fundal negru
    getch(); system("CLS"); system("Color 06");
    cout<<"Componentele sub diagonala secundara: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            if (i+j>n-1) cout<<" "<<M[i][j];
        } //cout<<endl; afisarea datelelor de pe fiecare rand
    }
    getch(); system("Color 07");//revenim la setarile initiale
}
```

Problema 5.2 - 13

Elaborați un cod C/C++ în care va construi în memorie și va afișa un array bidimensional cu elemente întregi pozitive, astfel încât pe diagonala principală să existe numai elemente egale cu 1, elementele de pe cele două „semi-diagonale” paralele cu diagonala principală și alăturate diagonalei principale să fie toate egale cu 2, elementele de pe următoarele „semidiagonale” să fie egale cu 3, etc.

Implementarea C++

```
#include <iostream>
#define n 5
#define m 5
using namespace std;
int i,j,M[n][m];
int main() {
    // diagonala principală este 1
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            M[i][j]=abs(i-j)+1;
        }
    }
    cout<<"Afișam componentele matricii M: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
}
```

Problema 5.2 - 14

Elaborați un cod C/C++ în care componentele a două matrici vor fi declarate implicit în cod. Afișați suma (sau diferența) acestor două matrici.

Implementarea C++

```
#include <iostream>
#define n 3 // linii
#define m 3 // coloane
using namespace std;
int i,j,C[20][20];
```

```

int main() {
    // declarăm matricea M cu componentele sale
    int M[n][m]={1,2,3},{5,1,7},{9,0,1}};
    // declarăm matricea N cu componentele sale
    int N[n][m]={2,3,4},{1,7,2},{0,1,2}};
    cout<<"Afisam componentele matricii M: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    cout<<"Afisam componentele matricii N: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<N[i][j];
        }
        cout<<endl;
    }
    cout<<"Afisam componentele matricii M+N: \n";
    for(i=0;i<n;i++){
        for (j=0;j<m;j++){
            C[i][j]=M[i][j]+N[i][j];
            cout<<" "<<C[i][j];
        }
        cout<<endl;
    }
}

```

Problema 5.2 - 15

Elaborați un cod C/C++ în care componentele a două matrici vor fi declarate implicit în cod. Afișați produsul acestor două matrici.

Implementarea C++

```

#include <iostream>
#define n 3 // linii
#define m 3 // coloane
using namespace std;
int i,j,k,C[20][20];
int main() {
    // declarăm matricea M cu componentele sale
    int M[n][m]={1,2,3},{5,6,7},{9,0,1}};

```

```

// declaram matricea N cu componentele sale
int N[n][m]={ {2,3,4},{6,7,8},{0,1,2}};
cout<<"Afisam componentele matricii M: \n";
for(i=0;i<n;i++){
    for(j=0;j<m;j++){
        cout<<" "<<M[i][j];
    }
    cout<<endl;
}
cout<<"Afisam componentele matricii N: \n";
for(i=0;i<n;i++){
    for(j=0;j<m;j++){
        cout<<" "<<N[i][j];
    }
    cout<<endl;
}
// calculam produsul M*N
for(i=0;i<n;i++){
    for (j=0;j<m;j++){
        C[i][j]=0;
        for(k=0;k<m;k++){
            C[i][j]+=M[i][k]*N[k][j];
        }
    }
}
cout<<"Afisam componentele matricii M*N: \n";
for(i=0;i<n;i++){
    for(j=0;j<m;j++){
        cout<<" "<<C[i][j];
    }
    cout<<endl;
}
}

```

Problema 5.2 – 16

Elaborați un cod C/C++ în care va copia elementele array-ului bidimensional, începând cu prima linie, într-un array unidimensional.

Implementarea C++

```

#include <iostream>
#define n 3
#define m 3

```



```

using namespace std;
int i,j,M[n][m]={1,2,3},{2,3,1},{3,1,2}},V[n*m];
int main() {
    cout<<"Afisam componentele matricii M: \n";
    for(i=0;i<n;i++){
        for(j=0;j<m;j++){
            cout<<" "<<M[i][j];
        }
        cout<<endl;
    }
    // transformam matricea in vector
    int k=0;
    for(i=0;i<n*m;i++){
        for(j=0;j<m;j++){
            V[k++]=M[i][j];
        }
    }
    cout<<"Afisam componentele vectorului V: \n";
    for(i=0;i<n*m;i++){
        cout<<" "<<V[i];
    }
}

```

5.3 Probleme propuse

1. Probleme de aplicare a array-urilor unidimensionale

- Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele unui array unidimensional de numere întregi:*
 - afișarea elementelor divizibile cu k (se va introduce de la tastatură);*
 - afișarea elementelor pare și divizibile cu k , dar nedivizibile cu t ;*
 - afișarea elementelor impare și divizibile cu k ;*
 - afișarea elementelor de pe poziții pare și nedivizibile cu k ;*
 - afișarea elementelor de pe poziții impare și nedivizibile cu k .*
- Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele unui array unidimensional de numere întregi cu n ($n < 100$) componente:*
 - calculați media aritmetică (armonică) a componentelor pare (impare);*
 - determinați componenta minimă (maximă) și poziția acesteia;*
 - determinați componenta minimă (maximă) și numărul ei de apariții;*
 - afișați toate numerele prime (sau cele din seria lui Fibonacci);*
 - afișați toate pătratele (cuburile) perfecte.*
- Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele unui array unidimensional de numere întregi cu n ($n < 100$) componente:*
 - determinați dacă pătratul (cubul) unei valori k aparține vectorului;*
 - să se înlocuiască fiecare componentă pară cu componenta minimă;*
 - să se determine suma (produsul) elementelor diferite de numărul k ;*
 - să se determine câte componente sunt egale cu a x -a componentă din vector și afișați aceste componente;*
 - să se determine câți indici sunt egali cu a x -a componentă din vector și afișați acești indici.*

4. *Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele unui array unidimensional de numere întregi cu n ($n < 100$) componente:*
- a) *să se insereze valoarea k pe poziția p ($p \leq n$), toate datele se citesc de la tastatură;*
 - b) *să se insereze între oricare două elemente media aritmetică a acestor două numere;*
 - c) *să se insereze după fiecare număr impar (par) pozitiv dublul (triplul) acestuia;*
 - d) *să se insereze după fiecare valoare egală cu 0 (sau oricare alt număr) din vector o valoare k introdusă de la tastatură.*
5. *Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele unui array unidimensional de numere întregi cu n ($n < 100$) componente:*
- a) *să se elimine valoarea de pe poziția p ;*
 - b) *să se elimine toate valorile egale cu x ;*
 - c) *să se mute la sfârșitul vectorului zerourile fără a schimba ordinea elementelor nenule;*
 - d) *să se elimine valorile pare (impare sau prime) din vector.*
6. *Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele unui array unidimensional de numere întregi cu n ($n < 100$) componente:*
- a) *să se inverseze ordinea componentelor astfel încât primul element devine ultimul, al doilea penultimul ș.a.m.d.*
 - b) *să se efectueze k permutări circulare la stânga (dreapta);*
 - c) *să se mute la sfârșitul unui vector toate elementele nule (nu se vor utiliza secvențe de ștergere);*
 - d) *să se rearanjeze elementele vectorului, astfel încât valorile pare să fie la început, iar cele impare să fie la sfârșit;*
 - e) *să se rearanjeze elementele vectorului, astfel încât valorile pozitive să fie la început, iar cele negative să fie la sfârșit.*

7. *Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele a două array-uri unidimensionale de numere întregi cu n ($n < 100$) componente fiecare:*
 - a) *se cere să se afișeze pe ecran numerele divizibile cu 5 care se găsesc doar în unul din cei doi vectori (sau în ambii vectori concomitent);*
 - b) *se cere să se afișeze la ecran, numerele care se găsesc doar în unul din cei doi vectori și nu sunt în celălalt;*
 - c) *se cere să se afișeze la ecran componentele comune ale vectorilor;*
 - d) *se cere să se afișeze la ecran numerele prime din cei doi vectori.*
8. *Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele a două mulțimi (array-uri unidimensionale) de numere reale cu n componente fiecare, dacă se știe că: $-1000 < n < 1000$.*
 - a) *să se afișeze un vector care memorează intersecția mulțimilor;*
 - b) *să se afișeze un vector care memorează reuniunea mulțimilor;*
 - c) *să se afișeze un vector care memorează diferența $(A-B)$;*
 - d) *să se afișeze un vector care memorează diferența $(B-A)$.*
9. *Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele unui array unidimensional de numere întregi cu n ($n < 100$) componente:*
 - a) *să se determine dacă elementele vectorului sunt distincte;*
 - b) *să se determine câte perechi (triplete) din vector sunt prime între ele;*
 - c) *să se determine câte perechi din vector au valori egale (pare);*
 - d) *să se determine câte perechi (triplete) din vector au ca sumă un număr prim cu cifra unităților egală cu un număr k ;*
 - e) *să se determine câte elemente conține cea mai lungă secvență de numere pare (impare) alăturate din vector;*
 - f) *să se afișeze la ecran lungimea maximă a unei secvențe din vector cu proprietatea că oricare două elemente aflate pe poziții consecutive sunt prime între ele.*

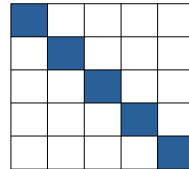
10. Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele a două array-uri unidimensionale de numere întregi cu n ($n < 100$) componente fiecare. Să se calculeze valoarea expresiilor:
- $E = (X[0] + Y[0]) * (X[1] + Y[1]) * \dots * (X[n-1] + Y[n-1]);$
 - $E = X[0] * Y[0] + X[1] * Y[1] + \dots + X[n-1] * Y[n-1];$
 - $E = \min(X[0], Y[0]) + \min(X[1], Y[1]) + \dots + \min(X[n-1], Y[n-1]);$
 - $E = \max(X[0], Y[0]) * \max(X[1], Y[1]) * \dots * \max(X[n-1], Y[n-1]).$
11. Elaborați un cod C/C++ care va prelucra următoarele operațiuni cu elementele a două array-uri unidimensionale de numere întregi cu n ($n < 100$) componente fiecare.
- a) să se determine suma a două polinoame, coeficienții celor două polinoame vor fi componentele celor doi vectori declarați;
 - b) să se determine diferența a două polinoame, coeficienții celor două polinoame vor fi componentele celor doi vectori declarați;
 - c) să se determine produsul a două polinoame, coeficienții celor două polinoame vor fi componentele celor doi vectori declarați;
 - d) să se determine câtul a două polinoame, coeficienții celor două polinoame vor fi componentele celor doi vectori declarați.
12. Elaborați un cod C/C++ care va genera într-un vector primele k numere pare (impare sau prime).
13. Elaborați un cod C/C++ care va genera într-un vector primele x numere pare divizibile cu k care au inversul un număr divizibil tot cu k .
14. Elaborați un cod C/C++ care va genera într-un vector primele x numere din șirul lui Fibonacci ($F[1]=1$, $F[2]=1$, $F[n]=F[n-1]+F[n-2]$, $n>2$).
- 15.³ Elaborați un cod C/C++ care va genera numere de telefon ce vor include doar numere întregi pozitive (0-9), respectând următorul format:
- a) toate numerele orange: 068XXXXXX; 069XXXXXX; 060XXXXXX;
 - b) toate numerele molcell: 078XXXXXX; 079XXXXXX; 070XXXXXX;
 - c) toate numerele unite: 067XXXXXX; 057XXXXXX; 047XXXXXX;

3 Pentru rezolvarea acestei probleme e necesar să aplicați elemente de combinatorică. Afișați toate numerele generate și stabiliți câte numere pot fi generate.

2. Probleme de aplicare a array-urilor bidimensionale

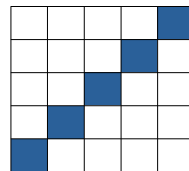
1. Elaborați un cod C/C++ care va prelucra operațiuni cu elementele din fiecare zonă necolorată a unui array bidimensional de forma $N \times N$, ce conține doar numere întregi:

- a) afișați elementul minim divizibil cu 3;
- b) afișați elementul maxim, pătrat perfect;
- c) afișați suma numerelor prime;
- d) produsul elementelor ce au un număr par de divizori.



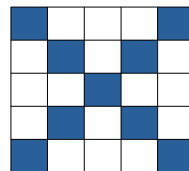
2. Elaborați un cod C/C++ care va prelucra operațiuni cu elementele din fiecare zonă necolorată a unui array bidimensional de forma $N \times N$, ce conține doar numere întregi:

- a) afișați elementul minim divizibil cu 5;
- b) afișați elementul maxim, cub perfect;
- c) afișați produsul numerelor prime;
- d) suma elementelor ce au cel puțin o cifră pară în poziția sa.



3. Elaborați un cod C/C++ care va prelucra operațiuni cu elementele din fiecare zonă necolorată a unui array bidimensional de forma $N \times N$, ce conține doar numere reale:

- a) afișați coordonatele elementul minim;
- b) afișați coordonatele elementul maxim;
- c) înlocuiți toate elementele ce au partea întreagă un număr par cu 0;
- d) înlocuiți toate elementele ce au partea întreagă un număr impar cu pătratul acestuia.



4. Elaborați un cod C/C++ care va forma un array unidimensional cu media aritmetică a elementelor aflate pe fiecare coloană a unei matrici cu n linii și m coloane, ce conține numere întregi pozitive.

5. <i>Elaborați un cod C/C++ care va construi în memorie o matrice pătratică cu elemente întregi pozitive, astfel încât pe ultima coloană și pe ultima linie a matricii să se afle numai elemente egale cu 1, iar oricare alt element al matricii să fie suma dintre elementul aflat imediat sub el și elementul aflat imediat în dreapta lui. Valoarea lui n ($n < 500$) se citește de la tastatură. Matrice se va afișa la ecran în mod standard: de la prima la ultima linie, elementele unei linii fiind scrise de la stânga la dreapta, cu spații între elementele oricărei linii.</i>	Exemplu: pentru $n=4$, se va afișa: <pre> 70 35 10 5 1 35 20 10 4 1 15 10 6 3 1 5 4 3 2 1 1 1 1 1 1 </pre>
6. <i>Elaborați un cod C/C++ care va construi în memorie o matrice pătratică cu elemente întregi pozitive, astfel încât pe prima coloană și pe prima linie a matricii să se afle numai elemente egale cu 2, iar oricare alt element al matricii să fie suma dintre elementul aflat imediat deasupra lui și elementul aflat imediat în stânga lui. Valoarea lui n ($n < 500$) se citește de la tastatură. Matrice se va afișa la ecran în mod standard: de la prima la ultima linie, elementele unei linii fiind scrise de la stânga la dreapta, cu spații între elementele oricărei linii.</i>	Exemplu: pentru $n=5$, se va afișa: <pre> 2 2 2 2 2 2 4 6 8 10 2 6 12 20 30 2 8 20 40 70 2 10 30 70 140 </pre>
7. <i>Elaborați un cod C/C++ care va construi în memorie o matrice pătratică formată numai din valori 1 și 2 astfel încât elementele de pe diagonala secundară și cea principală să fie egale cu 1, elementele situate între diagonalele matricii, în partea superioară și inferioară a acestora, să fie egale cu 1, iar restul elementelor din matrice să fie egale cu 2.</i>	Exemplu: pentru $n=5$, se va afișa: <pre> 1 1 1 1 1 2 1 1 1 2 2 2 1 2 2 2 1 1 1 2 1 1 1 1 1 </pre>

8. *Elaborați un cod C/C++ care va implementa rotirea elementelor unei matrici sub un unghi de 90° la dreapta (sau la stânga).*

9. Elaborați un cod C/C++ în care va fi declarată o matrice de dimensiunea NxM în mod implicit cu valori de 1 și 0. Se cere determinarea locurilor valorilor 1 care au ca vecini numărul 1 (vecinii trebuie să fie din cele 4 direcții: Nord, Sud, Vest și Est).

Exemplu:

fie o matrice 5x6:

```
1 0 1 1 1 1
0 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 0 1
1 1 1 1 1 1
```

10. Elaborați un cod C/C++ în care va fi declarată o matrice de dimensiunea NxM în mod implicit cu valori întregi pozitive (sau pot fi introduse de la tastatură). Matricea reprezintă un teren accidentat în urma căderii unor asteroizi de diverse dimensiuni. Fiecare element al matricii reprezintă altitudinea suprafeței de teren la care se referă acel element. Cineva aruncă o bilă pe acest teren, care cade în locul de coordonate (i,j). Să se determine dacă bila rămâne pe locul în care a căzut sau se va rostogoli în altă parte.

Exemplu:

fie o matrice 4x5:

```
12 3 4 56 6
1 5 3 3 7
22 2 4 7 3
89 32 45 3 7
```

dacă o bilă se va afla în [1][1], atunci ease va rostogoli în [1][0].

11. Elaborați un cod C/C++ în care va afișa mesajul decriptat prin culegerea caracterelor care corespund zerourilor din grilă când aceasta se suprapune peste textul cifrat. Oferindu-se un text cifrat într-o matrice de caractere și o grilă de aceleași dimensiuni, conținând numai elemente binare. Un model de problemă poate fi studiată în exemplul alăturat, unde textul decriptat va fi: „acest text”.

Exemplu:

fie o matrice 4x5:

```
a x 2 * c
% e t s z
t y w t ;
e z z x t
```

fie o grilă 4x5:

```
0 1 1 1 0
1 0 1 0 1
0 1 1 0 1
0 1 1 0 0
```

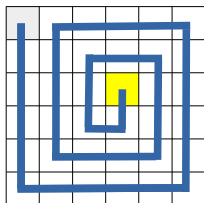
12. Elaborați un cod C/C++ care va implementa interschimbarea liniilor și coloanelor unei matrici pătrate, astfel încât în final elementele situate pe diagonala principală (sau secundară) să fie ordonate crescător.

13. Elaborați un cod C/C++ care va verifica dacă elementele de pe fiecare linie (sau coloană) sunt ordonate crescător (sau descrescător).

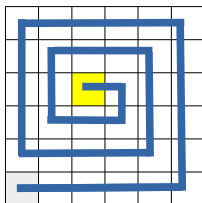
14. ***Se consideră două matrici pătratice cu componente reale (al doilea pătrat fiind mai mic decât primul). Se citește un număr natural p . Elaborați un cod C/C++ care va înlocui matricea a doua în prima matrice, începând cu poziția $[p][p]$, desigur dacă acest lucru este posibil.
15. ***Se consideră o matrice de elemente naturale. Se citește un număr natural n cu minimum 4 cifre. Elaborați un cod C/C++ care va completa fiecare linie a matricii cu cifrele reprezentării numărului n în baza b , unde b este numărul liniei+2.
16. ***Se citește o matrice de maximum 100 de elemente din mulțimea $\{0,1,2,\dots,9\}$, aranjate în maximum 10 linii a câte maximum 10 elemente fiecare. Elaborați un cod C/C++ care va verifica dacă fiecare linie este un palindrom (este același șir de cifre indiferent de ordinea de citire pe linie).
17. ***Se consideră o matrice cu maximum 100 de valori reale aranjate în maximum 10 linii de maximum 10 coloane. În fiecare linie sunt valori reprezentând înălțimile date în centimetri ale unor persoane din grupul dat de numărul liniei +1. Elaborați un cod C/C++ care va afișa:
 - a) grupul de persoane cu înălțimea mai mare ca o înălțime H ;
 - b) numărul de procente al persoanelor cu o înălțime mai mare decât H .
18. ***La o sesiune sport participă n persoane, numerotate de la 1 la n . Unele persoane cunosc alte persoane din sală, iar altele nu. Calculatorul care monitorizează reuniunea trebuie să comunice în final care este cea mai cunoscută persoană (adică este cunoscută de către cele mai multe persoane din sală). Pentru aceasta se introduc ca date în program perechi de numere de forma $[i][j]$ care semnifică dacă persoana i cunoaște persoana j (nu și invers). Elaborați un cod C/C++ care va aplica o matrice pătratică cu elemente din mulțimea $\{0,1\}$, să se afișeze numărul persoanei căutate.
19. ***Oferindu-se o matrice pătratică, cu elemente reale, ale cărei linii și coloane nu depășesc numărul 50. Elaborați un cod C/C++ care va afișa suma elementelor de pe diagonală I-a și suma elementelor de pe diagonală a II-a, calculate simultan (la o singură parcurgere a elementelor).
20. ***Oferindu-se o matrice, cu elemente reale. Elaborați un cod C/C++ care va determina dacă elementele din fiecare linie sunt ordonate ascendent.
21. Elaborați o secvență de cod C/C++ care va implementa următoarele parcurgeri și afișări ale elementelor unui array 2D conform imaginilor de la pagina următoare:

a) Parcurgerea sub formă de spirală (4 modalități din 8 posibile):

1	2	3	4	5	6
20	21	22	23	24	7
19	32	33	34	25	8
18	31	36	35	26	9
17	30	29	28	27	10
16	15	14	13	12	11



6	7	8	9	10	11
5	24	25	26	27	12
4	23	34	35	28	13
3	22	33	36	29	14
2	21	32	31	30	15
1	20	19	18	17	16



direcția: $NV \mapsto E$

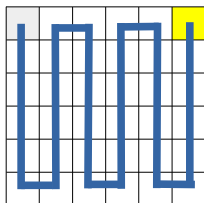
direcția: $NV \mapsto S$

direcția: $SV \mapsto N$

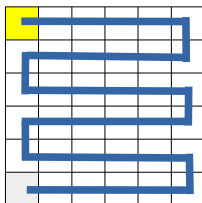
direcția: $SV \mapsto E$

b) Parcurgerea sub formă de șarpe (4 modalități din 8 posibile):

1	2	3	4	5	6
12	11	10	9	8	7
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30
36	35	34	33	32	31



6	7	18	19	30	31
5	8	17	20	29	32
4	9	16	21	28	33
3	10	15	22	27	34
2	11	14	23	26	35
1	12	13	24	25	36



direcția: $NV \mapsto SV$

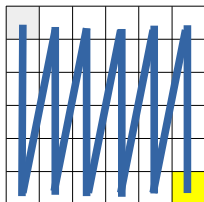
direcția: $NV \mapsto NE$

direcția: $SV \mapsto SE$

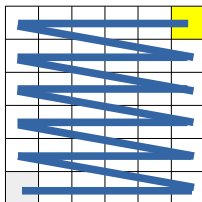
direcția: $SV \mapsto NV$

c) Parcurgerea sub formă de zig-zag (4 modalități din 8 posibile):

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36



6	12	18	24	30	36
5	11	17	23	29	35
4	10	16	22	28	34
3	9	15	21	27	33
2	8	14	20	26	32
1	7	13	19	25	31



direcția: $NV \mapsto SE$

direcția: $NV \mapsto NE$

direcția: $SV \mapsto NE$

direcția: $SV \mapsto SE$

Pentru fiecare formă de parcurgere prezentată, există următoarele modalități de parcurgere: $NV \mapsto E$, $NV \mapsto S$, $SV \mapsto N$, $SV \mapsto E$, $NE \mapsto V$, $NE \mapsto S$, $SE \mapsto N$ și $SE \mapsto V$. Dintre toate aceste modalități, mai sus au fost prezentate doar primele 4.

REZOLVAREA PROBLEMELOR CU STRING-URI ÎN LIMBAJUL C/C++

6.1 Aplicații elementare cu string-uri



- ✓ Unele probleme rezolvate vor fi însoțite de comentarii succinte.
- ✓ În acest compartiment vom aplica în mod implicit rezolvarea problemelor de diverse complexități, cu implementarea tipului de date caracter.
- ✓ Vom aplica un șir de caractere în stilul limbajului C, dar vom studia și biblioteca cstring.
- ✓ Verificați secvențele de cod C/C++ ale problemelor rezolvate, utilizați IDE-ul Code::Blocks.

Problema 6.1 - 01

Elaborați un cod C/C++ care va citi de la tastatură două cuvinte: numele și prenumele unei persoane. Afișați la ecran aceste informații. Șirul de caractere va fi prezentat în stilul limbajului C.

Implementarea C++

```
#include <iostream>
using namespace std;
char n[10],p[10]; // numele si prenumele
int main (){
    // Citirea datelor de la tastatura
    cout<<"Introdu numele persoanei: "; cin>>n;
    cout<<"Introdu prenumele persoanei: "; cin>>p;
    // Afișarea datelor la consola
    cout<<"\nAti introdus persoana: "<<n<<" "<<p<<endl;
}
```

Problema 6.1 - 02

Elaborați un cod C/C++ care va citi de la tastatură un set de maxim 10 simboluri. Implementați unele funcții implicite pentru lucrul cu caracterele. Șirul de caractere va fi prezentat în stilul limbajului C.

Implementarea C++

```
#include <iostream>
#include <cctype>
using namespace std;
char c[10]={'0','U','w','.','%','P','v'}; //simboluri
int main (){
    // Descrierea functiilor sunt prezentate in anexa 2
    cout<<"Aplicarea functiilor pentru caractere:\n"<<endl;
    cout<<"\tisalpha(c[1]): "<<isalpha(c[1]);
    cout<<" \tislower(c[2]): "<<islower(c[2])<<endl;
    cout<<"\tisupper(c[3]): "<<isupper(c[3]);
    cout<<" \tisdigit(c[4]): "<<isdigit(c[4])<<endl;
    cout<<"\ttolower(c[5]): "<<tolower(c[5]);
    cout<<"\ttoupper(c[6]): "<<toupper(c[6])<<endl;
    cout<<"\nSecventa de simboluri este: "<<<<endl;
}
```

Problema 6.1 - 03

Elaborați un cod C/C++ care va citi un cuvânt de la tastatură și va afișa fiecare caracter individual separat prin spațiu în ordine inversă. Șirul de caractere va fi prezentat în stilul limbajului C.

Implementarea C++

```
#include<iostream>
#include<cstring>
using namespace std;
char c[25]; int i,n;
int main(){
    cout<<"Introduceti cuvantul: "; cin>>c;
    cout<<"\nCaracterele cuvantului sunt: \n\t";
    i=strlen(c);
    while(i>=0){
        cout<<c[i]<<" "; i--;
    }
}
```

Problema 6.1 - 04

Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa numărul de cuvinte ale acestuia. Șirul de caractere va fi prezentat în stilul limbajului C.

Implementarea C++

```
#include<cstring>
using namespace std;
char mesaj[100]; int i=0,cuvinte=1;
int main(){
    cout<<"Introduceti mesajul: "; cin.get(mesaj,100);
    while(mesaj[i]!='\0'){
        /*verificam daca caracterul curent este spatiu, sau linie noua,
        sau caracter tab*/
        if(mesaj[i]==' ' || mesaj[i]=='\n' || mesaj[i]=='\t'){
            cuvinte++;
        }
        i++;
    }
    cout<<"Numarul total de cuvinte ale mesajului: "<<cuvinte;
}
```

Problema 6.1 - 05

Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa caracterul ce apare cel mai des în acest mesaj. Afișați de câte ori apare acesta. Șirul de caractere va fi prezentat în stilul limbajului C.

Implementarea C++

```
#include<iostream>
#include<cstring>
using namespace std;
char mesaj[100]; int c[255],i=0,m,ascii;
int main(){
    cout<<"Introduceti mesajul: "; cin.get(mesaj,100);
    // Setam frecventa tuturor caracterelor la 0
    for(i=0;i<255;i++){
        c[i] = 0;
    }
    //Citim frecventa fiecarui caracter
    i=0;
```

```

while(mesaj[i]!='\0'){
    ascii=(int)mesaj[i]; c[ascii] += 1;
    i++;
}
// Determinam care caracter se repeta cel mai des
m=0;
for(i=0;i<255;i++){
    if(i!=32){
        if(c[i]>c[m]) m = i;
    }
}
// Convertim codul ASCII al caracterului din integer in char
cout<<"Caracterul '"<<(char)m<<" apare de "<<c[m];
cout<<" ori in mesaj!";
}

```

Problema 6.1 - 06

Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va șterge toate aparițiile unui anumit cuvânt (sau expresie) din cadrul mesajului. Determinați de câte ori a fost șters acest cuvânt. Șirul de caractere va fi prezentat în stilul limbajului C (utilizați tipul de date pointer).

Implementarea C++

```

#include<iostream>
#include<cstring>
using namespace std;
char mesaj[100],c[25],*p; int n=0;
int main(){
    cout<<"Introduceti mesajul: "; cin.get(mesaj,100);
    cout<<"Introduceti cuvantul: "; cin>>c;
    // Eliminarea cuvantului din mesaj
    p=strstr(mesaj,c);
    while(p){
        // Stergem spatiul in plus dupa eliminarea cuvantului
        strcpy(p-1,p+strlen(c)); n++; p=strstr(p-1,c);
    }
    cout<<"\nCuvantul pentru stergere este: "<<c<<endl;
    cout<<"\nMesajul obtinut dupa eliminarea cuvantului: ";
    cout<<"\n\t"<<mesaj<<endl;
    cout<<"\nCuvantul \""<<c<<"\" a fost sters de "<<n<<" ori.";
}

```

Problema 6.1 – 07

Elaborați un cod C/C++ care va citi de la tastatură un șir de caractere format din maximum 10 caractere, reprezentând un număr real în care partea întreagă este separată de partea zecimală prin caracterul punct. Programul afișează pe ecran valoarea obținută prin adunarea numărului corespunzător părții întregi și a numărului corespunzător părții zecimale ale datei citite. Șirul de caractere va fi prezentat în stilul limbajului C (utilizați tipul de date pointer). Exemple: pentru 120.456, se va afișa 576; pentru 121.090, se va afișa 130.

Implementarea C++

```
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;
char numar[10],*c; int a,b;
int main(){
    cout<<"Introduceti un numar real: "; cin>>numar;
    //Separam partea intreaga de partea zecimala
    c=strchr(numar, '.');
    strncpy(numar,numar,c-numar);
    a=atol(numar); b=atol(c+1);
    while(b%10==0){
        b=b/10;
    }
    // Afisam aceste parti si suma lor
    cout<<"Suma partii reale si a celei fractionare: ";
    cout<<a<<" + "<<b<<" este "<<a+b;
}
```

Problema 6.1 - 08

Elaborați un cod C/C++ care va citi N cuvinte de la tastatură și le va aranja în ordine alfabetică. Șirul de caractere va fi prezentat în stilul limbajului C.

Implementarea C++

```
#include<iostream>
#include<cstring>
using namespace std;
char cuvant[50][25],aux[25]; int N,gasit=1,i;
int main(){
    cout<<"Introduceti numarul de cuvinte: "; cin>>N;
```

```

        for(i=0;i<N;i++){
            cout<<"cuvantul["<<i<<"]= "; cin>>cuvant[i];
        }
        // Sortarea cuvintelor conform metodei selectiei
        // Aceasta metoda poate fi studiata independent
        while(gasit==1){
            gasit=0;
            for(i=0;i<N-1;i++){
                if(strcmp(cuvant[i],cuvant[i+1]) > 0){
                    strcpy(aux, cuvant[i]);
                    strcpy(cuvant[i],cuvant[i+1]);
                    strcpy(cuvant[i+1],aux); gasit=1;
                }
            }
        }
        // Afisarea ascendenta
        cout<<"\nAfisarea alfabetica, de la A la Z: \n";
        for(i=0;i<N;i++){
            cout<<cuvant[i]<<endl;
        }
        // Afisarea descendenta
        cout<<"\nAfisarea alfabetica, de la Z la A: ";
        for(i=N;i>=0;--i){
            cout<<cuvant[i]<<endl;
        }
    }
}

```

Problema 6.1 - 09

Elaborați un cod C/C++ care va citi de la tastatură un șir de caractere și va afișa acest șir la ecran, repetându-l de N ori.

Implementarea C++

```

#include <iostream>
#include <cstring>
using namespace std;
char enunt[50]; int n;
int main() {
    // Citim un enunt de la tastatura, ca separator folosim spatiul
    cout<<"Introduceti o propozitie: "<<endl;
    cin.getline(enunt, 50);
    // Citim un numar de la tastatura
    cout<<"Introduceti un numar intreg pozitiv: "<<endl; cin>>n;
}

```



```

// Afisam enuntul de N ori conform conditiei
cout<<"\nPropozitia va fi afisata de "<<n<<" ori: "<<endl;
for (int i=0;i<n;i++){
    cout<<enunt<<endl;
}
}

```

Problema 6.1 - 10

Elaborați un cod C/C++ care va citi de la tastatură două șiruri de caractere. Afișați șirul obținut prin concatenarea ambelor șiruri inversate.

Implementarea C++

```

#include <iostream>
#include <cstring>
using namespace std;
char enunt1[50],enunt2[60];
int main() {
    // Citim primul enunt de la tastatura
    cout<<"Introduceti propozitia 1: "<<endl;
    cin.getline(enunt1, 50);
    // Citim al doilea enunt de la tastatura
    cout<<"Introduceti propozitia 2: "<<endl;
    cin.getline(enunt2, 60);
    // Afisam enuntul concatenat
    cout<<"\nConcatenarea celor doua enunturi inversate va fi:
"<<endl;
    cout<<strcat(strrev(enunt1),strrev(enunt2));
}

```

Problema 6.1 - 11

Elaborați un cod C/C++ care va prezenta un exemplu de aplicabilitate a instrucțiunilor ciclice de determinare a dimensiunii unui șir de caractere.

Implementarea C++

```

#include <iostream>
#include <string>
using namespace std;
char s1[50],s2[60]; int i=0;
int main() {
    // Citim trei siruri de la tastatura

```

```

cout<<"Introduceti sirul 1: "; cin.getline(s1, 50);
cout<<"Introduceti sirul 2: "; cin.getline(s2, 60);
// Metoda 1 cu instructiunea do...while
do{
    i++;
}
while(s1[i]!=0);
cout<<"\nDimensiunea sirui 1 va fi: "<<i;
// Metoda 2 cu instructiunea while
while(s2[i]!=0){
    i++;
}
cout<<"\nDimensiunea sirui 2 va fi: "<<i;
}

```

Problema 6.1 - 12

Elaborați un cod C/C++ care va prezenta un exemplu de aplicabilitate a funcțiilor de transformare a caracterelor (literelor) unui șir de caractere.

Implementarea C++

```

#include <iostream>
#include <cstring>
using namespace std;
char s1[50],s2[60]; int i=0,j=0;
int main() {
    // Citim doua siruri de la tastatura
    cout<<"Introduceti sirul 1: "; cin.getline(s1, 50);
    cout<<"Introduceti sirul 2: "; cin.getline(s2, 60);
    // Metoda 1
    cout<<"\nSirui 1 cu toate literele majuscule: "<<endl;
    for(i;i<strlen(s1);i++){
        s1[i]=toupper(s1[i]); // conversie in litere majuscule
        cout<<s1[i];
    }
    cout<<"\nSirui 2 cu toate literele minuscule: "<<endl;
    for(j;j<strlen(s2);j++){
        s2[j]=tolower(s2[j]); // conversie in litere minuscule
        cout<<s2[j];
    }
    // Metoda 2
    cout<<"\n\nSirui 1 cu toate literele majuscule: "<<endl;
    while(s1[i]!=0){

```

```

        if((s1[i]>=65) && (s1[i]<=90)) s1[i]=s1[i]+32; i++;
    }
    cout<<s1;
    cout<<"\nSirui 2 cu toate literele minuscule: "<<endl;
    while(s2[i]!=0){
        if((s2[i]>=97) && (s2[i]<=122)) s2[i]=s2[i]-32; i++;
    }
    cout<<s2;
}

```

Problema 6.1 - 13

Elaborați un cod C/C++ care va implementa 4 operații cu datele a două șiruri de caractere. Se vor implementa funcțiile `strlen()`, `strcpy()`, `strcmp()` și `strcat()`.

Implementarea C++

```

#include <iostream>
#include <cstring>
using namespace std;
char m1[30]="Matematica",m2[40]=" este regina stiintelor!",m3[30];
int main() {
    //Aplicarea functiei strlen()
    cout<<"Dimensiunea lui \"<<m1<<\" este: "<<strlen(m1)<<endl;
    //Aplicarea functiei strcpy()
    strcpy(m3,m1);
    cout<<"\nCopiem valoarea mesajului din m1 in m3: "<<endl;
    cout<<"Mesajul 1: "<<m1<<endl;
    cout<<"Mesajul 3: "<<m3<<endl;
    //Aplicarea functiei strcmp()
    cout<<"\nCompararea a doua mesaje: "<<endl;
    cout<<"Mesajul 1 & 2: "<<strcmp(m1,m2)<<endl;
    cout<<"Mesajul 1 & 3: "<<strcmp(m1,m3)<<endl;
    //Aplicarea functiei strcat()
    cout<<"\nConcatenarea a doua mesaje: "<<endl;
    strcat(m1,m2);
    cout<<"Primele doua mesaje unite: "<<m1<<endl;
}

```

Problema 6.1 - 14

Elaborați un cod C/C++ care va implementa compararea a două șiruri de caractere și determinați dacă ele rimează între ele.

Implementarea C++

```
#include <iostream>
#include <string>
using namespace std;
char c1[30],c2[40]; int i=0,j;
int main() {
    // Citim doua cuvinte de la tastatura
    cout<<"Introduceti cuvantul 1: "; cin.getline(c1, 30);
    cout<<"Introduceti cuvantul 2: "; cin.getline(c2, 40);
    cout<<"\nSolutia este: \n\t";
    // Verificarea existentei rimei
    while(c1[i]!=0){
        i++;
    }
    j=0;
    do{
        j++;
    }
    while(c2[j]!=0);
    // Afisarea rezultatului
    if((c1[i]==c2[j]) && (c1[i-1]==c2[j-1])){
        cout<<c1<<" rimeaza cu "<<c2;
    }
    else cout<<c1<<" nu rimeaza cu "<<c2;
}
```

Problema 6.1 – 15

Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa numărul: de vocale, de consoane, de cifre și de spații.

Implementarea C++

```
#include <iostream>
#include <cstring>
using namespace std;
char mesaj[50];
int v=0, c=0, cifre=0, spatii=0;
int main(){
    cout<<"Introdu un mesaj: "; cin.getline(mesaj, 50);
    for(int i=0; i<strlen(mesaj); ++i){
        // Numarul de vocale
        if(mesaj[i]=='a' || mesaj[i]=='e' || mesaj[i]=='i' ||
            mesaj[i]=='o' || mesaj[i]=='u' || mesaj[i]=='A' ||
```

```

        mesaj[i]=='E' || mesaj[i]=='I' || mesaj[i]=='O' ||
        mesaj[i]=='U'){
            ++v;
        }
        // Numarul de consoane
        else if((mesaj[i]>='a' && mesaj[i]<='z') || (mesaj[i]>='A' &&
            mesaj[i]<='Z')){
            ++c;
        }
        // Numarul de cifre de la 0 la 9
        else if(mesaj[i]>='0' && mesaj[i]<='9'){
            ++cifre;
        }
        // Numarul de spatii
        else if (mesaj[i]==' '){
            ++spatii;
        }
    }
    cout<<"\nVocale: \t"<<v; cout<<"\nConsoane: \t"<<c;
    cout<<"\nCifre: \t"<<cifre; cout<<"\nSpatii: \t"<<spatii;
}

```

6.2 Probleme propuse

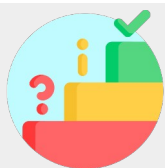
1. *Elaborați un cod C/C++ care va citi un mesaj (sau cuvânt) de la tastatură și va determina:*
 - a) *dacă acest cuvânt este palindrom;*
 - b) *dacă acest mesaj conține cel puțin V vocale;*
 - c) *dacă acest mesaj conține cel mult C consoane.*
2. *Elaborați un cod C/C++ care va citi un mesaj (sau cuvânt) de la tastatură și va elimina:*
 - a) *toate vocalele (sau toate consoanele) din mesaj;*
 - b) *toate cifrele (sau toate spațiile) din mesaj;*
 - c) *toți diftongii (ai, au, ei, eu, ii, iu, oi, ou, ui) din mesaj.*
3. *Elaborați un cod C/C++ care va citi două mesaje (sau cuvinte) de la tastatură și va afișa (sau va elimina):*
 - a) *toate vocalele (sau consoanele) comune;*
 - b) *toate numerele comune;*
 - c) *toți diftongii (ai, au, ei, eu, ii, iu, oi, ou, ui) comuni.*
4. *Elaborați un cod C/C++ care va citi două mesaje (sau cuvinte) de la tastatură și va afișa:*
 - a) *mesajul cu cele mai multe (sau mai puține) caractere;*
 - b) *mesajul obținut prin concatenarea acestor două mesaje;*
 - c) *fiecare mesaj după eliminarea cuvintelor comune lor.*
5. *Elaborați un cod C/C++ care va citi un mesaj (sau cuvânt) de la tastatură și va inversa acest mesaj (sau cuvânt).*
6. *Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa frecvența fiecărui caracter din acest mesaj.*
7. *Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va verifica dacă litera 'a' este prezentă în acest mesaj. Dacă este prezentă litera, atunci afișați toate pozițiile acesteia din mesaj.*

8. *Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va verifica dacă cuvântul „bravo” este prezent în acest mesaj. Dacă este prezent, atunci afișați toate pozițiile de început ale acestuia din mesaj.*
9. *Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa numărul de cuvinte ale acestui mesaj.*
10. *Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa toate anagramele și numărul acestora din mesaj.*
11. *Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa elementele din acest mesaj care conțin toate caracterele unui cuvânt declarat implicit în program.*
12. *Elaborați un cod C/C++ care va citi un mesaj (un număr întreg pozitiv) de la tastatură și va afișa codificarea lungimii acestui mesaj. Exemplu: pentru „1 111 233 333 3555” se va afișa „14213653”.*
13. *Elaborați un cod C/C++ care va citi un mesaj de la tastatură și va afișa prin spațiu toți indecșii de apariție a unui cuvânt declarat implicit în program. Exemplu: pentru mesajul „AABB ABCAB DCAB BCABB” și cuvântul „CAB” va afișa indecșii de început: 7 12 17.*
14. *Elaborați un cod C/C++ care să citească numele complet al unei persoane și să afișeze abrevierile prenumelui și prenumelui, cu excepția numelui de familie care este afișat așa cum este. Exemplu: dacă numele meu este „Andrian Emanuel Trofaila”, atunci rezultatul ar trebui să fie „A. E. Trofaila”.*
15. *Elaborați un cod C/C++ care va citi un mesaj (sau cuvânt) de la tastatură și va afișa toate permutările, toate aranjamentelor și toate combinațiilor de caractere ale acestui mesaj (sau cuvânt).*
16. *Elaborați un cod C/C++ care va citi un mesaj (sau cuvânt) de la tastatură și va afișa criptarea acestuia astfel:*
 - a) înlocuirea caracterelor cu codurile ASCII corespunzătoare în baza 10 (sau în baza 16);
 - b) înlocuirea vocalelor cu cea mai apropiată consoană din dreapta ($a \rightarrow b$, $e \rightarrow f$, $i \rightarrow j$, $o \rightarrow p$, $u \rightarrow v$), iar aceste consoane cu numere ($b \rightarrow 2$, $f \rightarrow 6$, $j \rightarrow 10$, $p \rightarrow 16$, $v \rightarrow 22$);
 - c) înlocuirea literelor cu poziția acestora în alfabetul englez: $a \rightarrow 1$, $b \rightarrow 2$, $c \rightarrow 3$, ..., $z \rightarrow 26$.

4 Această problemă este opțională. Soluționarea acesteia se poate realiza cu ajutorul subprogramelor, această temă va fi abordată la cursul „Programare procedurală”.

STUDIUL INDIVIDUAL GHIDAT DE PROFESOR

7.1 Lucrarea nr.1 – Probleme elementare



- ✓ Fiecare elev va realiza sarcinile strict conform condiției. Rezultatele obținute vor fi prezentate profesorului sub formă de raport.
- ✓ Fiecare elev va citi atent condițiile problemei individuale și va schița schema bloc a problemei.
- ✓ Elaborarea secvenței de cod C/C++ în mediul de programare Code::Blocks.

Problema 1

Pentru literele C, H și O, fiecare elev va introduce corespunzător următoarele date: C = data nașterii, H = luna nașterii și O = numărul de ordine din registrul grupei. Datele se vor defini implicit în program.

Un cercetător studiază un compus chimic descoperit pe planeta X230Pi, în urma analizelor efectuate, a dedus că o moleculă din acest compus este formată din C atomi de carbon, O atomi de oxigen și H atomi de hidrogen. Știind că masa atomului de carbon este 12, masa atomului de oxigen este 16, iar masa atomului de hidrogen este 1, să se scrie un program care să calculeze și să afișeze masa moleculară a acestui compus.

Exemplu:

- Dacă eu sunt născut la data de 22. 07 și mă aflu în registrul grupei al 15-lea pe listă, atunci pentru mine voi avea următoarele date: C=22, H=7 și O=15, obținem formula compusului: $C_{22}H_7O_{15}$.

Problema 2

Pentru litera R și N , fiecare elev va introduce corespunzător următoarele date: R = data și luna nașterii, unde luna nașterii reprezintă partea zecimală și N = numărul de ordine din registrul grupei modulo 10. Datele se vor defini implicit în program.

Un matematician trebuie să determine aria unui poligon regulat S , în dependență de raza cercului circumscris lui R . Se cunoaște că R este un număr real, ce conține maxim două zecimale, iar poligonul este format din N laturi.

Exemplu:

- Dacă eu sunt născut la data de 16.03 și mă aflu în registrul grupei al 28-lea pe listă, atunci pentru mine voi avea următoarele date: $R=16.03$ și $N=28\%10$, deci $N=8$, obținem un poligon regulat cu 8 laturi.

Problema 3

Pentru literele v_1 , v_2 și t , fiecare elev va introduce corespunzător următoarele date: v_1 = data și luna nașterii, unde data indică numărul de zecimale; v_2 = data și luna nașterii elevului imediat următor din registrul grupei, ultimul elev din listă va primi ca valori pentru v_2 datele primei persoane din registrul grupei și t = numărul de ordine din registrul grupei a persoanei. Datele se vor defini implicit în program.

Un sportiv aleargă cu o viteză constantă de v_1 m/s. După un timp t , apare un al doilea sportiv pe traseul de cursă, ce menține o viteză constantă de v_2 m/s. Elaborați un program care să afișeze după câte secunde al doilea sportiv ajunge în rând cu primul. Dacă acesta nu-l poate ajunge, se va afișa valoarea -1.

Exemplu:

- Dacă eu sunt născut la data de 28.11 și mă aflu în registrul grupei al 6-lea pe listă, iar persoana imediat următoare mie este născută la data de 08.01 atunci pentru mine voi avea următoarele date: $v_1=11.28$ m/s, $v_2=1.08$ m/s și $t=6$ min.

7.2 Lucrarea nr. 2 – Structuri de control



- ✓ Fiecare elev va realiza sarcinile strict conform condiției. Rezultatele obținute vor fi prezentate profesorului sub formă de raport.
- ✓ Fiecare elev va citi atent condițiile problemei individuale și va schița schema bloc a problemei.
- ✓ Elaborarea secvenței de cod C/C++ în mediul de programare Code::Blocks.

1. Instrucțiuni de decizie și de salt

Pentru litera A, fiecare elev va introduce numărul de ordine personal corespunzător registrului grupei, iar pentru valoarea x - luna nașterii sale. Elevii care sunt născuți într-o lună pară, vor introduce numărul cu „-”. Elaborați cele trei programe în limbajul C/C++ pentru a implementa instrucțiunile de decizie: IF și SWITCH, precum și instrucțiunile opționale: GOTO și RETURN.

Să se calculeze valoarea funcției $f(x)=y$, în punctul x citit de la tastatură, unde funcția are următoarea formă:

$$f(x)=\begin{cases} 2x^4-A, & \text{dacă } x < -5 \\ x+\sqrt{A}, & \text{dacă } -5 \leq x < -2 \\ \frac{x+A}{x(A-x)}, & \text{dacă } -2 \leq x < 8 \\ -4x^3+A, & \text{dacă } x \geq 8 \end{cases}$$

Exemple:

- Dacă eu mă aflu în registrul grupei al 15-lea și sunt născut la data de 22.05, atunci pentru mine voi avea următoarele date: A=15 și x=5.
- Dacă eu mă aflu în registrul grupei al 8-lea și sunt născut la data de 12.08, atunci pentru mine voi avea următoarele date: A=8 și x=-8.

2. Instrucțiuni de ciclare și de salt

Elaborați cele trei programe în limbajul C/C++ pentru a implementa instrucțiunile de ciclare: FOR, WHILE și DO ... WHILE.

Elaborați câte un program pentru sumarea primilor $N=100$ termeni ai seriilor următoare și verificați numeric egalitățile date, astfel încât să nu fie luat în considerație termenul sumei al X -lea. Termenul al X -lea este numărul de ordine personal din registrul grupei.

Exemplu:

- Dacă eu mă aflu în registrul grupei al 27-lea, atunci eu voi calcula suma seriei astfel încât să nu fie inclus în sumă termenul al 27-lea.

Nr.	Condiția problemei pentru fiecare elev din subgrupă
01	$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \frac{1}{11} + \frac{1}{12} + \dots = +\infty$
02	$\frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \frac{1}{7} - \frac{1}{8} + \frac{1}{9} - \frac{1}{10} + \frac{1}{11} - \frac{1}{12} + \dots = \ln(2)$
03	$\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \frac{1}{17} - \frac{1}{19} + \frac{1}{21} - \frac{1}{23} + \dots = \frac{\pi}{4}$
04	$\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \frac{1}{6^2} + \frac{1}{7^2} + \frac{1}{8^2} + \frac{1}{9^2} + \frac{1}{10^2} + \frac{1}{11^2} + \frac{1}{12^2} + \dots = \frac{\pi^2}{6}$
05	$\frac{1}{1^4} + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \frac{1}{5^4} + \frac{1}{6^4} + \frac{1}{7^4} + \frac{1}{8^4} + \frac{1}{9^4} + \frac{1}{10^4} + \frac{1}{11^4} + \frac{1}{12^4} + \dots = \frac{\pi^4}{90}$
06	$1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \frac{1}{7!} + \frac{1}{8!} + \frac{1}{9!} + \frac{1}{10!} + \frac{1}{11!} + \frac{1}{12!} + \dots = e$
07	$\frac{1}{2 \cdot 3 \cdot 4} - \frac{1}{4 \cdot 5 \cdot 6} + \frac{1}{6 \cdot 7 \cdot 8} - \frac{1}{8 \cdot 9 \cdot 10} + \frac{1}{10 \cdot 11 \cdot 12} - \frac{1}{12 \cdot 13 \cdot 14} + \dots = \frac{\pi - 3}{4}$
08	$\sum_{i=0}^{\infty} \frac{1}{16^i} \left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6} \right) = \pi$

Elaborați câte un program pentru calcularea produsului primilor $N=100$ factori ai următoarelor produse infinite și verificați numeric egalitățile date, astfel încât să nu fie luat în considerație factorul al Y -lea. Factorul al Y -lea este numărul de ordine personal din registrul grupei.

Exemplu:

- Dacă eu mă aflu în registrul grupei al 21-lea, atunci eu voi calcula produsul astfel încât să nu fie inclus factorul al 21-lea.

Nr.	Condiția problemei pentru fiecare elev din subgrupă
09	$\frac{2^2-1}{2^2+1} \cdot \frac{3^2-1}{3^2+1} \cdot \frac{4^2-1}{4^2+1} \cdot \frac{5^2-1}{5^2+1} \cdot \frac{6^2-1}{6^2+1} \cdot \frac{7^2-1}{7^2+1} \cdot \frac{8^2-1}{8^2+1} \cdot \frac{9^2-1}{9^2+1} \cdot \dots = \frac{\pi}{\sinh(\pi)}$
10	$\frac{2^3-1}{2^3+1} \cdot \frac{3^3-1}{3^3+1} \cdot \frac{4^3-1}{4^3+1} \cdot \frac{5^3-1}{5^3+1} \cdot \frac{6^3-1}{6^3+1} \cdot \frac{7^3-1}{7^3+1} \cdot \frac{8^3-1}{8^3+1} \cdot \frac{9^3-1}{9^3+1} \cdot \dots = \frac{2}{3}$
11	$\left(1 - \frac{1}{2^2}\right) \cdot \left(1 - \frac{1}{3^2}\right) \cdot \left(1 - \frac{1}{4^2}\right) \cdot \left(1 - \frac{1}{5^2}\right) \cdot \left(1 - \frac{1}{6^2}\right) \cdot \left(1 - \frac{1}{7^2}\right) \cdot \left(1 - \frac{1}{8^2}\right) \cdot \dots = \frac{1}{2}$
12	$\left(1 + \frac{1}{2^2}\right) \cdot \left(1 + \frac{1}{3^2}\right) \cdot \left(1 + \frac{1}{4^2}\right) \cdot \left(1 + \frac{1}{5^2}\right) \cdot \left(1 + \frac{1}{6^2}\right) \cdot \left(1 + \frac{1}{7^2}\right) \cdot \left(1 + \frac{1}{8^2}\right) \cdot \dots = \frac{\sinh(\pi)}{2\pi}$
13	$\left(1 - \frac{1}{2^3}\right) \cdot \left(1 - \frac{1}{3^3}\right) \cdot \left(1 - \frac{1}{4^3}\right) \cdot \left(1 - \frac{1}{5^3}\right) \cdot \left(1 - \frac{1}{6^3}\right) \cdot \left(1 - \frac{1}{7^3}\right) \cdot \dots = \frac{\cosh\left(\frac{\pi\sqrt{3}}{2}\right)}{3\pi}$
14	$\left(1 + \frac{1}{2^3}\right) \cdot \left(1 + \frac{1}{3^3}\right) \cdot \left(1 + \frac{1}{4^3}\right) \cdot \left(1 + \frac{1}{5^3}\right) \cdot \left(1 + \frac{1}{6^3}\right) \cdot \left(1 + \frac{1}{7^3}\right) \cdot \dots = \frac{\cosh\left(\frac{\pi\sqrt{3}}{2}\right)}{2\pi}$
15	$\frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{6}{5} \cdot \frac{6}{5} \cdot \frac{8}{7} \cdot \frac{8}{7} \cdot \frac{10}{9} \cdot \frac{10}{9} \cdot \frac{12}{11} \cdot \frac{12}{11} \cdot \frac{14}{13} \cdot \frac{14}{13} \cdot \frac{16}{15} \cdot \frac{16}{15} \cdot \frac{18}{17} \cdot \dots = 2$
16	$\frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2}}}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{2}}}}}}{2} \cdot \dots = 2$



- ✓ Afișarea rezultatului trebuie să fie realizată cu minim 5 cifre zecimale pentru fiecare dintre sumele sau produsele prezentate mai sus.
- ✓ Implementați instrucțiunea **CONTINUE** în mod corespunzător cerinței problemei individuale.
- ✓ Termenul sau factorul ce trebuie să fie eliminat din calcule să fie definit implicit în program, se va utiliza directiva **# define**.

7.3 Lucrarea nr. 3 – Array & String



- ✓ Fiecare elev va realiza sarcinile strict conform condiției. Rezultatele obținute vor fi prezentate profesorului sub formă de raport.
- ✓ Fiecare elev va citi atent condițiile problemei individuale și va elabora secvența de cod C/C++ în mediul de programare Code::Blocks.

1. Array unidimensional (1D)

Pentru litera A, fiecare elev va introduce numărul de ordine personal corespunzător registrului grupei, astfel pentru $A=10$, $X[A]$ va conține $A+7=17$ elemente. Elaborați un program în limbajul C/C++ pentru a rezolva următoarea problemă. Numărul A trebuie să fie definit implicit în program.

Nr.	Condiția problemei pentru fiecare elev din subgrupă
01	Fie un array unidimensional $X[A+7]$. Determinați suma și cantitatea elementelor pare divizibile cu 3.
02	Fie un array unidimensional $X[A+7]$. Determinați media aritmetică și produsul elementelor de pe poziții impare.
03	Fie un array unidimensional $X[A+70]$. Comparați produsul elementelor pozitive cu suma elementelor pare.
04	Fie un array unidimensional $X[A+7]$. Determinați elementul maximal al masivului și de aflat dacă el este pozitiv sau negativ.
05	Fie un array unidimensional $X[A+7]$. Calculați diferența dintre produsul elementelor și media lor aritmetică. Determinați paritatea diferenței.
06	Fie un array unidimensional $X[A+7]$. Determinați produsul și cantitatea elementelor pozitive prime din masiv.
07	Fie un array unidimensional $X[A+7]$. Determinați cantitatea și media aritmetică a elementelor divizibile la trei.

Nr.	Condiția problemei pentru fiecare elev din subgrupă
08	Fie un array unidimensional $X[A+7]$. Determinați produsul sumelor elementelor pare și celor impare din masiv
09	Fie un array unidimensional $X[A+7]$. Determinați produsul dintre cantitatea elementelor divizibile la doi și suma lor.
10	Fie un array unidimensional $X[A+7]$. Determinați media aritmetica a elementelor impare din prima $3/4$ din masiv.
11	Fie un array unidimensional $X[A+7]$. Determinați suma dintre primul element al masivului și ultimul element impar al masivului.
12	Fie un array unidimensional $X[A+7]$. Determinați suma și cantitatea elementelor negative din prima jumătate a masivului.
13	Fie un array unidimensional $X[A+7]$. Determinați cantitatea elementelor impare printre elementele pozitive ale masivului.
14	Fie un array unidimensional $X[A+7]$. Comparați suma elementelor din prima jumătate a masivului cu media aritmetică a elementelor din a doua jumătate a masivului.
15	Fie un array unidimensional $X[A+7]$. Determinați elementul minimal printre elementele cu poziții pare ale masivului.
16	Fie un array unidimensional $X[A+7]$. Comparați suma elementelor pare cu produsul elementelor negative, în afară de primul și ultimul element.

2. Array bidimensional (2D)

Fie un array bidimensional $X[N][N]$, unde N este un număr par pozitiv. O matrice pătratică $X[N][N]$ o vom diviza în 4 zone conform imaginii alăturate.

Elaborați un program în limbajul C/C++ pentru a rezolva următoare problemă cu array-uri bidimensionale (masive).

Zona 1	Zona 2
Zona 4	Zona 3

Nr.	Condiția problemei pentru fiecare elev din subgrupă
01	<i>Schimbați cu locul elementele primului rând cu elementele ultimei coloane.</i>
02	<i>Aranjați în ordine crescătoare elementele fiecărui rând (sau verificați dacă sunt elementele în ordine crescătoare).</i>
03	<i>Schimbați cu locul elementele zonei I cu elementele zonei IV.</i>
04	<i>Schimbați cu locul elementul maximal al masivului cu elementul minimal al masivului.</i>
05	<i>Schimbați cu locul elementele penultimului rând cu elementele rândului indicat de utilizator.</i>
06	<i>Aranjați în ordine crescătoare elementele fiecărei coloane (sau verificați dacă sunt elementele în ordine crescătoare).</i>
07	<i>Schimbați cu locul elementele zonei II cu elementele zonei III.</i>
08	<i>Schimbați cu locul elementul maximal al rândului unu cu elementul minimal al ultimei coloane.</i>
09	<i>Schimbați cu locul elementele coloanei doi cu elementele coloanei indicate de utilizator.</i>
10	<i>Aranjați în ordine descrescătoare elementele fiecărui rând (sau verificați dacă sunt elementele în ordine descrescătoare).</i>
11	<i>Schimbați cu locul elementele zonei III cu elementele zonei IV.</i>
12	<i>Schimbați cu locul elementul minimal al ultimei coloane cu elementul maximal de pe rândul indicat de utilizator.</i>
13	<i>Schimbați cu locul elementele primei coloane cu elementele ultimului rând.</i>
14	<i>Aranjați în ordine descrescătoare elementele fiecărei coloane (sau verificați dacă sunt elementele în ordine descrescătoare).</i>
15	<i>Schimbați cu locul elementele zonei II cu elementele zonei IV.</i>
16	<i>Schimbați cu locul elementul maximal al coloanei indicate de utilizator cu elementul minimal de pe rândul trei.</i>

3. String (șiruri de caractere)

Fie următorul șir de caractere: „Buna ziua, ma numesc A, sunt nascut la data de D in L. Mamea mea este M si tatal meu este T. Disciplina mea preferata este X. ” Fiecare elev va completa șirul de caractere cu datele sale personale conform exemplului de mai jos.

Exemplu de date de intrare:

➤ A= „Andrian Ciobanu”, D= „22 mai 2002”, L= „orasul Balti, comuna Elizabetha”, M= „Liudmila Ciobanu”, T= „Eugen Ciobanu” si X= „educatia fizica”.

Elaborați un program în limbajul C/C++ pentru a rezolva următoare problemă cu tipul de date string (șir de caractere).

Nr.	Condiția problemei pentru fiecare elev din grupă
01	De comparat cantitatea vocalelor cu cantitatea consoanelor din șir.
02	De determinat de câte ori în șir se întâlnește diftongii: ai, au, ei, eu, ii, iu, oi, ou și ui.
03	De șters din șir toate zerourile și de înlocuit pe acestea cu un spațiu.
04	De aranjat caracterele variabilei A, M și T în ordine alfabetică.
05	De aflat de câte ori este întâlnit în șir fiecare din caractere.
06	De șters din șir toate vocalele de pe poziții pare.
07	De șters din șir toți diftongii: ai, au, ei, eu, ii, iu, oi, ou și ui.
08	De determinat valorile și pozițiile ultimelor două vocale din șir.
09	De schimbat cu locul caracterele unu cu trei, cinci cu șapte, nouă cu unsprezece ș.a.m.d. Dacă ultimul caracter va fi fără pereche, el va rămâne neschimbat.
10	De indicat simbolurile care se întâlnesc o singură dată în text.
11	De inversat șirul, folosind principiile clasice de inversare a masivelor.
12	De șters simbolul “ , ” din șir și de calculat numărul de înlăturări.

Nr.	Condiția problemei pentru fiecare elev din grupă
13	<i>De determinat valorile și pozițiile primelor trei consoane din șir.</i>
14	<i>Din textul dat de șters caracterele majuscule și de calculat lungimea textului rămas.</i>
15	<i>De șters din șir caracterele de pe poziții impare.</i>
16	<i>De șters din șir caracterele minuscule de pe poziții impare.</i>
17	<i>De schimbat cu locul primul și ultimul caracter din fiecare cuvânt al șirului.</i>
18	<i>De efectuat căutarea perechilor de cuvinte, în care unul din cuvinte este inversa altuia.</i>
19	<i>De aranjat cuvintele din șir în ordine alfabetică.</i>
20	<i>De determinat cuvintele cu caracterele aranjate în ordine alfabetică.</i>
21	<i>De inversat cuvintele de pe poziții impare.</i>
22	<i>De șters litera din mijloc din toate cuvintele de lungime impară.</i>
23	<i>De aranjat cuvintele din șir în ordine inversă.</i>
24	<i>De determinat cuvântul care conține numărul maximal de consoane.</i>
25	<i>De șters din șir cuvintele ce se întâlnesc o singură dată.</i>
26	<i>De schimbat cu locul cel mai scurt cuvânt cu cel mai lung cuvânt.</i>
27	<i>De șters din șir cuvintele ce se repetă.</i>
28	<i>De aranjat cuvintele șirului în ordine descrescătoare după cantitatea de vocale din cuvinte.</i>
29	<i>De determinat cuvintele ce se întâlnesc în șir de k ori. Numărul k va fi indicat de utilizator.</i>
30	<i>De găsit cuvintele simetrice de lungime pară.</i>
31	<i>De șters din șir cuvintele simetrice.</i>
32	<i>De inversat cuvintele de lungime pară ce se află pe poziții impare.</i>

7.4 Model de lucru pentru studiul individual

Studiul individual nr.1

Model de foaie de titlu a lucrării

Studiu individual nr.1 la unitatea de curs

Programare structurată

cu genericul

Elemente de bază ale limbajului de programare C/C++

Profesor NUME PRENUME

Nume Prenume	Grupa	Link-ul de la secvența video	Nota

Unități de conținut propuse:

1. Interfața mediului de programare CodeBlocks v.20;
2. Vocabularul și sintaxa limbajului C/C++;
3. Operații de intrare-ieșire.

Modalități de evaluare a lucrării individuale:

1. Comunicare sincronă și asincronă;
2. Demonstrare funcționalității produsului program pe un dispozitiv electronic de calcul: smartPhone, smartTab sau notebook (sau PC);
3. Crearea unei secvențe video în care elevul explică pas cu pas realizarea tuturor sarcinilor propuse, deci pentru fiecare studiu individual elevul va trebui să realizeze câte o secvență video corespunzătoare și să aplice în comunicarea sa limbajul de specialitate corespunzător unității de conținut;
4. În lucrarea finală fiecare elev va trimite și link-ul cu secvența video realizată conform sarcinii propuse.

Problema 1.1

Un cercetător studiază un compus chimic descoperit pe planeta BC1M23. În urma analizelor efectuate, a dedus că o moleculă din acest compus este formată din atomi de carbon, atomi de hidrogen și că acest compus posibil să fie o formă de alchenă. Știind formula moleculară generală a alchenelor și că masa atomului de carbon este 12, iar masa atomului de hidrogen este 1, să se scrie un program care să calculeze și să afișeze masa moleculară relativă a acestui compus chimic.

Rezolvare:

Formula moleculară generală a alchenelor este C_nH_{2n} , cu valori întregi succesive începând cu valoarea 2, astfel se obține seria omoloagă a alchenelor. Fie sunt un elev ce în registrul grupei de elevi mă aflu pe poziția 8, prin urmare eu voi obține conformula substanței chimice: C_8H_{16} , adică denumirea chimică este Caprylene. Masa moleculară relativă a acestui compus chimic va fi: $M(C_8H_{16}) = 8 \cdot 12.011 + 16 \cdot 1.0079 = 96.088 + 16.1264 = 112.2144 \text{ g/mol}$.

Listingul programului în limbajul C/C++

```
#include <iostream>
using namespace std;
#define N 8
#define MC 12.011
#define MH 1.0079
int main(){
    cout<<"Problema nr. 1"<<endl;
    cout<<"Am obtinut formula moleculara C8H16."<<endl;
    cout<<"Masa moleculara relativa pentru C8H16 este ";
    cout<<N*MC+2*N*MH<<" g/mol."<<endl;
}
```

Rezultatele execuției programului în mediul de programare CodeBlocks:

```
Problema nr. 1
Am obtinut formula moleculara C8H16.
Masa moleculara relativa pentru C8H16 este 112.214 g/mol.
```

Problema 1.2

Un matematician trebuie să determine aria unui poligon regulat S , în dependență de raza cercului circumscris lui R . Se cunoaște că R este un număr real, ce conține maxim două zecimale, iar poligonul este format din N laturi.

Rezolvare:

Formulele necesare pentru soluționarea acestei probleme sunt:

Formula 1	Formula 2	Formula 3	Formula 4
$S = \frac{1}{4} \cdot n \cdot a^2 \cdot \cot\left(\frac{\pi}{n}\right)$	$P = n \cdot a$	$R = \frac{a}{2 \cdot \sin\left(\frac{\pi}{n}\right)}$	$r = \frac{a}{2 \cdot \tan\left(\frac{\pi}{n}\right)}$

Fie sunt un elev ce în registrul grupei de elevi mă aflu pe poziția 8, iar 8 modulo 10 va fi 8, deci $N=8$, iar data de naștere este 22 iunie, deci $R=22.06$. Pentru a afla aria poligonului nostru este necesară lungimea laturii care o determinăm conform formulei 3 din tabelul de mai sus: $a = 2 \cdot R \cdot \sin\left(\frac{\pi}{n}\right)$.

Acum vom determina lungimea laturii poligonului regulat:

$$a = 2 \cdot R \cdot \sin\left(\frac{\pi}{n}\right) = 2 \cdot 22.06 \cdot \sin\left(\frac{\pi}{8}\right) = 44.12 \cdot 0.38268343236 = 16.8839930357.$$

Acum vom determina aria poligonului regulat:

$$S = \frac{1}{4} \cdot n \cdot a^2 \cdot \cot\left(\frac{\pi}{n}\right) = \frac{1}{4} \cdot n \cdot \left(2 \cdot R \cdot \sin\left(\frac{\pi}{n}\right)\right)^2 \cdot \cot\left(\frac{\pi}{n}\right) = n \cdot R^2 \cdot \left(\sin\left(\frac{\pi}{n}\right)\right)^2 \cdot \cot\left(\frac{\pi}{n}\right).$$

Cu ajutorul calculelor aproximative vom obține:

$$S = 8 \cdot 22.06^2 \cdot \left(\sin\left(\frac{\pi}{8}\right)\right)^2 \cdot \cot\left(\frac{\pi}{8}\right) = 3893.1488 \cdot 0.1464466094 \cdot 2.41421356.$$

Deci aria poligonului regulat conform datelor inițiale: $S = 1376.4359569$.

Listingul programului în limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
#define N 8
#define PI 3.14159265359
```

```
#define R 22.06
float a;
int main(){
    cout<<"Problema nr. 2"<<endl;
    cout<<"Am obtinut un poligon cu 8 laturi."<<endl;
    cout<<"Lungimea laturii poligonului regulat este ";
    a=2*R*sin(PI/N);
    cout<<a<<" u.l."<<endl;
    cout<<"Aria poligonului regulat este ";
    cout<<0.25*N*pow(a,2)*(1/tan(PI/N))<<" u.p."<<endl;
}
```

Rezultatele execuției programului în mediul de programare CodeBlocks:

```
Problema nr. 2
Am obtinut un poligon cu 8 laturi.
Lungimea laturii poligonului regulat este 16.884 u.l.
Aria poligonului regulat este 1376.44 u.p.
```

Problema 1.3

Doi bicicliști pleacă în același timp: X din orașul A , iar Y din orașul B . Primul se deplasează cu o viteză medie de v_1 km/oră, iar celălalt are o viteză cu 3 km/oră mai mare. Distanța dintre cele două orașe este S km. După cât timp T și la ce distanță D de orașul A se vor întâlni?

Rezolvare:

Fie sunt un elev ce în registrul grupei de elevi mă aflu pe poziția 8 dintr-o listă de 32 de elevi, iar $8 \cdot 32 = 256$, deci $S = 256$ km, iar data de naștere este 22 iunie, deci $v_1 = 22.06$ km/h, iar $v_2 = v_1 + 3 = 25.06$ km/h.

- Putem afla distanța pe care o parcurg ambii bicicliști timp de o oră: $d_1 = v_1 + v_2$, $v = 22.06 + 25.06 = 47.12$ (km).
- Cunoșcând distanța totală și distanța parcursă de cei doi bicicliști împreună într-o oră, putem afla timpul scurs de la plecare până la întâlnire: $T = S : d_1$, $T = 256 : 47.12 = 5.43293718$ (h).

- Știind că X, plecând din A merge timp de ≈ 5.43 ore cu viteza de 22.06 km/oră, putem afla la ce distanță de A s-a produs întâlnirea: $D=T \cdot v_1$, $D=5.43293 \cdot 22.06=119.850594$ (km).

Prin urmare formulele pentru determinarea lui D și T sunt:

$$T = \frac{S}{v_1 + v_2} (h) \quad \text{și} \quad D = \frac{S \cdot v_1}{v_1 + v_2} (km) \quad \text{sau} \quad D = T \cdot v_1 (km).$$

Listingul programului în limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
#define S 256
#define v1 22.06
float v2,T,D;
int main(){
    cout<<"Problema nr. 3"<<endl;
    cout<<"Dupa cat timp se vor intalni biciclistii?"<<endl;
    v2=v1+3;
    T=S/(v1+v2);
    cout<<"\tRaspuns: T= "<<T<<" ore."<<endl;
    cout<<"La ce distanta fata de A se vor intalni ei?"<<endl;
    D=T*v1;
    cout<<"\tRaspuns: D= "<<D<<" km."<<endl;
}
```

Rezultatele execuției programului în mediul de programare CodeBlocks:

```
Problema nr. 3
Dupa cat timp se vor intalni biciclistii?
Raspuns: T= 5.43294 ore.
La ce distanta fata de A se vor intalni ei?
Raspuns: D= 119.851 km.
```

Model de foaie de titlu a lucrării

Studiul individual nr.2 la unitatea de curs

Programare structurată

cu genericul

Structuri de control ale limbajului de programare C/C++

Profesor NUME PRENUME

Nume Prenume	Grupa	Link-ul de la secvența video	Nota

Unități de conținut propuse:

1. Instrucțiuni de decizie: IF și SWITCH;
2. Instrucțiuni ciclice: FOR, WHILE și DO...WHILE;
3. Instrucțiuni de salt: BREAK, CONTINUE și GOTO (la dorință).

Modalități de evaluare a lucrării individuale:

1. Comunicare sincronă și asincronă;
2. Demonstrare funcționalității produsului program pe un dispozitiv electronic de calcul: smartPhone, smartTab sau notebook (sau PC);
3. Crearea unei secvențe video în care elevul explică pas cu pas realizarea tuturor sarcinilor propuse, deci pentru fiecare studiu individual elevul va trebui să realizeze câte o secvență video corespunzătoare și să aplice în comunicarea sa limbajul de specialitate corespunzător unității de conținut;
4. În lucrarea finală fiecare elev va trimite și link-ul cu secvența video realizată conform sarcinii propuse.

Problema 2.1

Să se calculeze valoarea funcției $f(x)=y$, în punctul x citit de la tastatură, unde funcția are următoarea formă:

$$g(x)=\begin{cases} 2x^4-A, & \text{dacă } x < -5 \\ x+\sqrt{A}, & \text{dacă } -5 \leq x < -2 \\ -5x-\sqrt{A}+x^A, & \text{dacă } x \geq -2 \end{cases}$$

Rezolvare:

Fie sunt un elev ce în registrul grupei de elevi mă aflu pe poziția 8 dintr-o listă de 32 de elevi, prin urmare $A=8$. Vom analiza cele 3 cazuri conform funcției de mai sus:

1. pentru $x < -5$, fie $x = -6$, prin urmare $g(-6) = 2 \cdot (-6)^4 - 8$, deci $g(-6) = 2 \cdot 1296 - 8 = 2592 - 8 = 2584$;
2. pentru $-5 \leq x < -2$, fie $x = -3$, prin urmare $g(-3) = -3 + \sqrt{8}$, deci $g(-3) = -3 + 2.82842712474619 = -0.1715728752538099$;
3. pentru $x \geq -2$, fie $x = 2$, prin urmare $g(2) = (-5) \cdot 2 - \sqrt{8} + 2^8$, deci $g(2) = -10 - 2.82842712474619 + 256 = 243.1715728752538$.

Listingul programului în limbajul C/C++ (instrucțiunea IF)

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
#define A 8
int x;
int main(){
    cout<<"Problema nr. 1 - Instructiunea IF"<<endl;
    cout<<"Introducem valoarea lui x, x= "; cin>>x;
    cout<<"Valoarea functiei g("<<x<<"")= ";
    if (x<-5) cout<<fixed<<setprecision(2)<<2*pow(x,4)-A;
    else if (x>=-5 && x<-2)
    cout<<fixed<<setprecision(3)<<x+sqrt(A);
    else cout<<fixed<<setprecision(4)<<-5*x-sqrt(A)+pow(x,A);
}
```


Listingul programului în limbajul C/C++ (instrucțiunea SWITCH)

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
#define A 8
int x,k=0;
int main(){
    cout<<"Problema nr. 1 - Instrucțiunea SWITCH"<<endl;
    cout<<"Introducem valoarea lui x, x= "; cin>>x;
    cout<<"Valoarea funcției g("<<x<<"= ";
    // Aplicam operatorul conditional
    k=(x<-5)?(k=k+1):(x>=-5 && x<-2)?(k=k+2):(k=k+3);
    switch(k){
        case 1:cout<<fixed<<setprecision(2)<<2*pow(x,4)-A;break;
        case 2:cout<<fixed<<setprecision(3)<<x+sqrt(A);break;
        case 3:cout<<fixed<<setprecision(4)<<5*x-sqrt(A+pow(x,A));
    break;
    }
}
```

Listingul programului în limbajul C/C++ (instrucțiunea GOTO)

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
#define A 8
int x;
int main(){
    cout<<"Problema nr. 1 - Instrucțiunea GOTO"<<endl;
    cout<<"Introducem valoarea lui x, x= "; cin>>x;
    cout<<"Valoarea funcției g("<<x<<"= ";
    if (x<-5) goto g1;
    else if (x>=-5 && x<-2) goto g2;
    else goto g3;
    g1: {
        cout<<fixed<<setprecision(2)<<2*pow(x,4)-A;
        return 1;}
    g2: {
```

```

        cout<<fixed<<setprecision(3)<<x+sqrt(A);
        return 1;}
g3: {
        cout<<fixed<<setprecision(4)<<-5*x-sqrt(A)+pow(x,A);
        return 1;}
}

```

Rezultatele execuției programelor în mediul de programare CodeBlocks:

Rezultatele pentru cazul 1:

Problema nr. 1 - Instrucțiunea IF
Introducem valoarea lui x, x= -6
Valoarea funcției g(-6)= 2584.00

Rezultatele pentru cazul 2:

Problema nr. 1 - Instrucțiunea SWITCH
Introducem valoarea lui x, x= -3
Valoarea funcției g(-3)= -0.172

Rezultatele pentru cazul 3:

Problema nr. 1 - Instrucțiunea GOTO
Introducem valoarea lui x, x= 2
Valoarea funcției g(2)= 243.1716

Problema 2.2

Elaborați câte un program pentru sumarea primilor N=100 termeni ai seriei următoare și verificați numeric egalitățile date, astfel încât să nu fie luat în considerație termenul sumei al X-lea. Termenul al X-lea este numărul de ordine personal din registrul grupei. Fie seria numerică:

$$\sum_{n=1}^{100} (1^2+2^2+3^2+4^2+5^2+6^2+7^2+8^2+9^2+10^2+11^2+12^2+13^2+14^2+15^2+...+n^2)$$

Rezolvare:

Fie sunt un elev ce în registrul grupei de elevi mă aflu pe poziția 8 dintr-o listă de 32 de elevi, prin urmare trebuie să fie exclus termenul $X=8$ din sumă. Să presupunem că avem doar 10 termeni în loc de 100, prin urmare suma mea va avea următoarea formă, cu condiția că voi exclude termenul al 8-lea:

$$\sum_{n=1}^{10} (1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 9^2 + 10^2) = 1 + 4 + 9 + 16 + 25 + 36 + 49 + 81 + 100.$$

Prin urmare $\sum_{n=1}^{10} (1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 9^2 + 10^2) = 321.$

Listingul programului în limbajul C/C++ (instrucțiunea FOR)

```
#include <iostream>
#include <cmath>
using namespace std;
#define n 10
#define x 8
int main(){
    int i,S=0;
    cout<<"Problema nr. 2 - Instrucțiunea FOR"<<endl;
    cout<<"Din suma vom omite termenul al "<<x<<"-lea."<<endl;
    cout<<"S("<<n<<","<<x<<")= ";
    for (i=1; i<=n;i++){
        if (i==x) continue;
        S+=pow(i,2);
        cout<<pow(i,2)<<"+";
    }
    cout<<"0."<<endl;
    cout<<"S("<<n<<","<<x<<")= "<<S<<endl;
}
```

Listingul programului în limbajul C/C++ (instrucțiunea WHILE)

```
#include <iostream>
#include <cmath>
using namespace std;
#define n 10
```

```

#define x 8
int main(){
    int i=1,S=0;
    cout<<"Problema nr. 2 - Instructiunea WHILE"<<endl;
    cout<<"Din suma vom omite termenul al "<<x<<"-lea."<<endl;
    cout<<"S("<<n<<","<<x<<")= ";
    while(i<=n){
        S+=pow(i,2);
        cout<<pow(i,2)<<"+";
        i++;
    }
    cout<<"(-"<<pow(x,2)<<")."<<endl;
    cout<<"S("<<n<<","<<x<<")= "<<S-pow(x,2)<<endl;
}

```

Listingul programului în limbajul C/C++ (instrucțiunea DO...WHILE)

```

#include <iostream>
#include <cmath>
using namespace std;
#define n 10
#define x 8
int main(){
    int i=1,S=0;
    cout<<"Problema nr. 2 - Instructiunea DO...WHILE"<<endl;
    cout<<"Din suma vom omite termenul al "<<x<<"-lea."<<endl;
    cout<<"S("<<n<<","<<x<<")= ";
    do{
        S+=pow(i,2);
        cout<<pow(i,2)<<"+";
        i++;
    }
    while(i<=n);
    cout<<"(-"<<pow(x,2)<<")."<<endl;
    cout<<"S("<<n<<","<<x<<")= "<<S-pow(x,2)<<endl;
}

```

Rezultatele execuției programelor în mediul de programare CodeBlocks:

Rezultatele pentru cazul 1:

```
Problema nr. 2 - Instrucțiunea FOR
Din suma vom omite termenul al 8-lea.
S(10,8)= 1+4+9+16+25+36+49+81+100+0.
S(10,8)= 321
```

Rezultatele pentru cazul 2:

```
Problema nr. 2 - Instrucțiunea WHILE
Din suma vom omite termenul al 8-lea.
S(10,8)= 1+4+9+16+25+36+49+64+81+100+(-64).
S(10,8)= 321
```

Rezultatele pentru cazul 3:

```
Problema nr. 2 - Instrucțiunea DO...WHILE
Din suma vom omite termenul al 8-lea.
S(10,8)= 1+4+9+16+25+36+49+64+81+100+(-64).
S(10,8)= 321
```

Vă propun mai jos un model de program care include toate cele 3 bucle: FOR, WHILE și DO...WHILE. Această secvență de cod include un meniu și implementarea unor particularități ale instrucțiunilor ciclice:

```
#include <iostream>
#include <cmath>
using namespace std;
#define n 10
#define x 8
int main(){
    int i,S=0,var;
    cout<<"Problema nr. 2"<<endl;
    cout<<"Din suma vom omite termenul al "<<x<<"-lea."<<endl;
    // MENIUL APLICATIEI
    cout<<"Alege o optiune: "<<endl;
    cout<<"\t1 - Instrucțiunea FOR"<<endl;
    cout<<"\t2 - Instrucțiunea WHILE"<<endl;
    cout<<"\t3 - Instrucțiunea DO...WHILE"<<endl;
    cout<<"Aleg instrucțiunea: "; cin>>var;
```

```

switch (var){
    case 1: goto i1;
    case 2: goto i2;
    case 3: goto i3;
}
// MENIUL ETICHETELOR
i1:{
    cout<<"\tInstructiunea FOR"<<endl;
    cout<<"\tS("<<n<<","<<x<<")= ";
    for (i=1; i<=n;i++){
        if (i==x) continue;
        S+=pow(i,2); cout<<pow(i,2)<<"+";
    }
    cout<<"0."<<endl;
    cout<<"\tS("<<n<<","<<x<<")= "<<S<<endl;
    return 1;
}
i2:{
    cout<<"\tInstructiunea WHILE"<<endl;
    cout<<"\tS("<<n<<","<<x<<")= ";
    i=1;
    while(i<=n){
        S+=pow(i,2); cout<<pow(i,2)<<"+"; i++;
    }
    cout<<"(-"<<pow(x,2)<<")."<<endl;
    cout<<"\tS("<<n<<","<<x<<")= "<<S-pow(x,2)<<endl;
    return 1;
}
i3:{
    cout<<"\tInstructiunea DO...WHILE"<<endl;
    cout<<"\tS("<<n<<","<<x<<")= ";
    i=1;
    do{
        S+=pow(i,2); cout<<pow(i,2)<<"+"; i++;
    }
    while(i<=n);
    cout<<"(-"<<pow(x,2)<<")."<<endl;
    cout<<"\tS("<<n<<","<<x<<")= "<<S-pow(x,2)<<endl;
    return 1;
}
}

```

Model de foaie de titlu a lucrării

Studiul individual nr.3 la unitatea de curs
Programare structurată
cu genericul
Tipurile de date ARRAY și STRING în C/C++

Profesor NUME PRENUME

Nume Prenume	Grupa	Link-ul de la secvența video	Nota

Unități de conținut propuse:

- 1. Tipul de date array unidimensional (sau array 1D);*
- 2. Tipul de date array bidimensional (sau array 2D);*
- 3. Tipul de date string.*

Modalități de evaluare a lucrării individuale:

- 1. Comunicare sincronă și asincronă;*
- 2. Demonstrare funcționalității produsului program pe un dispozitiv electronic de calcul: smartPhone, smartTab sau notebook (sau PC);*
- 3. Crearea unei secvențe video în care elevul explică pas cu pas realizarea tuturor sarcinilor propuse, deci pentru fiecare studiu individual elevul va trebui să realizeze câte o secvență video corespunzătoare și să aplice în comunicarea sa limbajul de specialitate corespunzător unității de conținut;*
- 4. În lucrarea finală fiecare elev va trimite și link-ul cu secvența video realizată conform sarcinii propuse.*

Problema 3.1

Fie un array unidimensional $X[A+7]$. Determinați suma și cantitatea elementelor pare, pozitive și nedivizibile cu 5.

Rezolvare:

Fie sunt un elev ce în registrul grupei de elevi mă aflu pe poziția 8 dintr-o listă de 32 de elevi, prin urmare array-ul meu unidimensional X va conține $8+7=15$ elemente. Să presupunem că avem următorul array unidimensional:

$X = \{23, 13, -24, -92, 45, 71, 43, -23, 99, 24, -92, 76, 20, 95, 88\}$

Prin urmare vom nota prin M – mulțimea elementelor pare, pozitive și nedivizibile cu 5: $M = \{24, 76, 88\}$, suma acestor elemente este: $S = 188$.

Vă propun mai jos un model de program care include toate cele 3 bucle: FOR, WHILE și DO...WHILE. Această secvență de cod include un meniu și implementarea unor particularități ale instrucțiunilor ciclice aplicate pentru tipul de date array:

Listingul programului în limbajul C/C++

```
#include <iostream>
#include <cmath>
using namespace std;
int main(){
    // Datele initiale pentru array-ul unidimensional
    int X[]={23,13,-24,-92,45,71,43,-23,99,24,-92,76,20,95,88};
    int n=sizeof X/sizeof X[0];
    int i,s=0,k=0,var;
    // MENIUL APLICATIEI
    cout<<"Alege o optiune: "<<endl;
    cout<<"\t1 - Instructiunea FOR"<<endl;
    cout<<"\t2 - Instructiunea WHILE"<<endl;
    cout<<"\t3 - Instructiunea DO...WHILE"<<endl;
    cout<<"Aleg instructiunea: "; cin>>var;
    switch (var){
        case 1: goto i1;
        case 2: goto i2;
        case 3: goto i3;
    }
}
```



```

// MENIUL ETICHETELOR
i1:{
    cout<<"\tInstructiunea FOR"<<endl;
    for(i=0;i<n;i++){
        if(X[i]%2==0 && X[i]>0 && X[i]%5!=0){
            s+=X[i]; k++;
        }
    }
    cout<<"\tConform conditiei vom avea "<<k<<" elemente.";
    cout<<"\n\tAfisam suma conform coditiei: \t"<<s;
    return 1;
}
i2:{
    cout<<"\tInstructiunea WHILE"<<endl;
    i=0;
    while(i<n){
        if(X[i]%2==0 && X[i]>0 && X[i]%5!=0){
            s+=X[i]; k++;
        }
        i++;
    }
    cout<<"\tConform conditiei vom avea "<<k<<" elemente.";
    cout<<"\n\tAfisam suma conform coditiei: \t"<<s;
    return 1;
}
i3:{
    cout<<"\tInstructiunea DO...WHILE"<<endl;
    i=0;
    do{
        if(X[i]%2==0 && X[i]>0 && X[i]%5!=0){
            s+=X[i]; k++;
        }
        i++;
    }
    while(i<n);
    cout<<"\tConform conditiei vom avea "<<k<<" elemente.";
    cout<<"\n\tAfisam suma conform coditiei: \t"<<s;
    return 1;
}
}

```

Rezultatele execuției programelor în mediul de programare CodeBlocks:

Rezultatele pentru cazul 1:

```
Alege o optiune:  
  1 - Instructiunea FOR  
  2 - Instructiunea WHILE  
  3 - Instructiunea DO...WHILE  
Aleg instructiunea: 1  
  Instructiunea FOR  
  Parcurgem componentele si determinam suma!  
  Conform conditiei vom avea doar 3 elemente.  
  Afisam suma conform coditiei:  188
```

Rezultatele pentru cazul 2:

```
Alege o optiune:  
  1 - Instructiunea FOR  
  2 - Instructiunea WHILE  
  3 - Instructiunea DO...WHILE  
Aleg instructiunea: 2  
  Instructiunea WHILE  
  Parcurgem componentele si determinam suma!  
  Conform conditiei vom avea doar 3 elemente.  
  Afisam suma conform coditiei:  188
```

Rezultatele pentru cazul 3:

```
Alege o optiune:  
  1 - Instructiunea FOR  
  2 - Instructiunea WHILE  
  3 - Instructiunea DO...WHILE  
Aleg instructiunea: 3  
  Instructiunea DO...WHILE  
  Parcurgem componentele si determinam suma!  
  Conform conditiei vom avea doar 3 elemente.  
  Afisam suma conform coditiei:  188
```

În loc de instrucțiunea goto se poate de aplicat instrucțiunea switch.

Problema 3.2

Fie un array bidimensional $X[N][N]$, unde N este un număr par pozitiv. Matricea $X[N][N]$ o vom diviza în 4 zone conform imaginii alăturate. Elaborați un program în limbajul C/C++ pentru a afișa suma elementelor din zona 1 și produsul elementelor impare mai mici decât A din zona 3.

Zona 1	Zona 2
Zona 4	Zona 3

Rezolvare:

Fie sunt un elev ce în registrul grupei de elevi mă aflu pe poziția 8 dintr-o listă de 32 de elevi, prin urmare pentru varianta mea $A=8$. Să presupunem că avem următorul array bidimensional:

$$X = \begin{pmatrix} 12 & 21 & 14 & -9 \\ 34 & -1 & 83 & 63 \\ 43 & 62 & -3 & 17 \\ -2 & 20 & 39 & -7 \end{pmatrix}$$

Vom avea următoarele array-uri bidimensionale $Z1$ și $Z3$ cu elementele:

$$Z1 = \begin{pmatrix} 12 & 21 \\ 34 & -1 \end{pmatrix} \quad \text{și} \quad Z3 = \begin{pmatrix} -3 & 17 \\ 39 & -7 \end{pmatrix}.$$

Așadar, suma elementelor din $Z1$ este $S(Z1)=66$, iar produsul elementelor impare mai mici decât $A=8$ este $P(Z3,A)=21$.

Vă propun mai jos un model de program care include toate cele 3 bucle: FOR, WHILE și DO...WHILE. Această secvență de cod include un meniu și implementarea unor particularități ale instrucțiunilor ciclice aplicate pentru tipul de date array:

Listingul programului în limbajul C/C++

```
#include <iostream>
#define n 4
#define A 8
using namespace std;
int main(){
    int M[n][n]={
        {12,21,14,-9},{34,-1,83,63},{43,62,-3,17},{-2,20,39,-7}
    };
    int i,j,s=0,p=1,var;
    // AFISAREA ARRAY-ULUI BIDIMENSIOANL
    cout<<"Afisam componentele matriciei: \n";
```

```

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        cout<<" "<<M[i][j];
    }
    cout<<endl;
}
// MENIUL APLICATIEI
cout<<"Alege o optiune: "<<endl;
cout<<"\t1 - Instructiunea FOR"<<endl;
cout<<"\t2 - Instructiunea WHILE"<<endl;
cout<<"\t3 - Instructiunea DO...WHILE"<<endl;
cout<<"Aleg instructiunea: "; cin>>var;
switch (var){
    case 1: goto i1;
    case 2: goto i2;
    case 3: goto i3;
}
// MENIUL ETICHETELOR
i1:{
    cout<<"\tInstructiunea FOR"<<endl;
    for(i=0;i<n/2;i++){
        for(j=0;j<n/2;j++){
            s+=M[i][j];
        }
    }
    cout<<"\tSuma elementelor din Z1 este "<<s<<endl;
    for(i=n/2;i<=n;i++){
        for(j=n/2;j<=n;j++){
            if(M[i][j]%2!=0 && M[i][j]<A){
                p*=M[i][j];
            }
        }
    }
    cout<<"\tProdusul elementelor din Z3 este "<<p<<endl;
    return 1;
}
i2:{
    cout<<"\tInstructiunea WHILE"<<endl;
    i=0;
    while(i<n/2){
        j=0;
        while(j<n/2){

```

```

        s+=M[i][j]; j++;
    }
    i++;
}
cout<<"\tSuma elementelor din Z1 este "<<s<<endl;
i=n/2;
while(i<=n){
    j=n/2;
    while(j<=n){
        if(M[i][j]%2!=0 && M[i][j]<A){
            p*=M[i][j];
        }
        j++;
    }
    i++;
}
cout<<"\tProdusul elementelor din Z3 este "<<p<<endl;
return 1;
}
i3:{
    cout<<"\tInstructiunea DO...WHILE"<<endl;
    i=0;
    do{
        j=0;
        while(j<n/2){
            s+=M[i][j]; j++;
        }
        i++;
    }
    while(i<n/2);
    cout<<"\tSuma elementelor din Z1 este "<<s<<endl;
    i=n/2;
    do{
        j=n/2;
        do{
            if(M[i][j]%2!=0 && M[i][j]<A){
                p*=M[i][j];
            }
            j++;
        }
        while(j<=n); i++;
    }
}

```

```

        while(i<=n);
        cout<<"\tProdusul elementelor din Z3 este "<<p<<endl;
        return 1;
    }
}

```

Rezultatele execuției programelor în mediul de programare CodeBlocks:

Rezultatele pentru cazul 1:

```

Afisam componentele matricii:
12 21 14 -9
34 -1 83 63
43 62 -3 17
-2 20 39 -7
Alege o optiune:
    1 - Instructiunea FOR
    2 - Instructiunea WHILE
    3 - Instructiunea DO...WHILE
Aleg instructiunea: 1
    Instructiunea FOR
    Suma elementelor din Z1 este 66
    Produsul elementelor din Z3 este 21

```

Rezultatele pentru cazul 2:

```

Aleg instructiunea: 2
    Instructiunea WHILE
    Suma elementelor din Z1 este 66
    Produsul elementelor din Z3 este 21

```

Rezultatele pentru cazul 3:

```

Aleg instructiunea: 3
    Instructiunea DO...WHILE
    Suma elementelor din Z1 este 66
    Produsul elementelor din Z3 este 21

```

În loc de instrucțiunea goto se poate de aplicat instrucțiunea switch.

Problema 3.3

Fie următorul șir de caractere: „**Buna ziua, ma numesc A, sunt nascut la data de D in L. Mama mea este M si tatal meu este T. Disciplina mea preferata este X.**” Fiecare elev va completa șirul de caractere cu datele sale personale. Elaborați un program în limbajul C/C++ pentru a rezolva următoarea problemă cu tipul de date string (șir de caractere):

1. de afișat codul ASCII pentru toate vocalele mari;
2. de transformat toate literele „a” în litera „e” și invers;
3. de șters toate consoanele mici din șir;
4. de numărat câte cifre conține șirul și de contorizat de câte ori se repetă fiecare;
5. de determinat ce zi din săptămână a fost când m-am născut.

Rezolvare:

Fie sunt un elev ce dețin următoarele informații: A= „Alexei Ciobanu”, D= „24 mai 2009”, L= „orasul Balti, comuna Elizaveta”, M= „Liudmila Ciobanu”, T= „Eugen Ciobanu” si X= „Informatica”. Mesajul nostru va fi: Buna ziua, ma numesc Alexei Ciobanu, sunt nascut la data de 24 mai 2009 in orasul Balti, comuna Elizaveta. Mama mea este Liudmila Ciobanu si tatal meu este Eugen Ciobanu. Disciplina mea preferata este Informatica.

Pentru început vom realiza condiția 1:

Codurile ASCII pentru vocalele mici sunt:

A=065; E=069; I=073; O=079; U=085;

Mesajul final va fi:

Buna ziua, ma numesc 065lexei Ciobanu, sunt nascut la data de 24 mai 2009 in orasul Balti, comuna 069lizaveta. Mama mea este Liudmila Ciobanu si tatal meu este 069ugen Ciobanu. Disciplina mea preferata este 073nformatica.

Continuăm cu realizarea condiției 2:

Mesajul final va fi:

Bune ziue, me numasc Alaxai Ciobenu, sunt nescut le dete de 24 mei 2009 in oresul Belti, comune Elizevate. Meme mae asta Liudmile Ciobenu si tetel mau este Eugan Ciobenu. Discipline mae prafarete este Informetice.

Continuăm cu realizarea condiției 3:

Mesajul final va fi:

Bua iua, a ue Aeei Cioau, u au a aa e 24 ai 2009 i oau Bai, oua Eiaea. Maa ea este Liuia Cioau i aa eu ee Eue Cioau. Diiia ea eaaa ee Ioaia.

Continuăm cu realizarea condiției 4:

Mesajul final compus doar din numere va fi: 24 2009.

Numărul total de cifre: 6.

1. Cifre de 0 avem 2.

3. Cifre de 4 avem 1.

2. Cifre de 2 avem 2.

4. Cifre de 9 avem 1.

Continuăm cu realizarea condiției 5:

Eu m-am nascut la date de 24.5.2009. Atunci a fost a 6-a zi din săptămâna. (0=Duminica, 1=Luni, 2=Marți, 3=Miercuri.

Listingul programului în limbajul C/C++

```
#include <iostream>
#include <cstring>
#define z 24    // variabila zi(1..31)
#define l 5     // variabila luna(1..12)
#define a 2009 // variabila anul
using namespace std;
int main(){
    int i,var,num=0,ziua;
    char mesaj[100]="Buna ziua, parintii mei sunt Alexei si Eliza-
veta. Azi, 24 mai 2009 este ziua mea de nastere.";
    // MENIUL APLICATIEI
    cout<<"Alege o optiune: "<<endl;
    cout<<"\t1 - Realizarea conditiei 1"<<endl;
    cout<<"\t2 - Realizarea conditiei 2"<<endl;
    cout<<"\t3 - Realizarea conditiei 3"<<endl;
    cout<<"\t4 - Realizarea conditiei 4"<<endl;
    cout<<"\t5 - Realizarea conditiei 5"<<endl;
    cout<<"Aleg instructiunea: "; cin>>var;
    switch (var){
        case 1: goto cond1;
        case 2: goto cond2;
```



```

        case 3: goto cond3;
        case 4: goto cond4;
        case 5: goto cond5;
    }
    // Am realizat conditia 1
    cond1:{
        cout<<"\tSolutia conditiei 1: \n\t";
        for (i=0;i<strlen(mesaj);i++){
            if(strchr("AEIOU",mesaj[i])){
                cout<<(int) mesaj[i];
            }
            else cout<<mesaj[i];
        }
        return 1;
    }
    // Am realizat conditia 2
    cond2:{
        cout<<"\tSolutia conditiei 2: \n\t";
        for (i=0;i<strlen(mesaj);i++){
            if(mesaj[i]=='a') mesaj[i]='e';
            else if(mesaj[i]=='e') mesaj[i]='a';
            else continue;
        }
        cout<<mesaj; return 1;
    }
    // Am realizat conditia 3
    cond3:{
        cout<<"\tSolutia conditiei 3: \n\t";
        for (i=0;i<strlen(mesaj);i++){
            while(strchr("bcdfhjklmnpqrstvwxyz",mesaj[i])){
                for (int j=i; j<=strlen(mesaj); j++){
                    mesaj[j]=mesaj[j+1];
                }
            }
        }
        cout<<mesaj; return 1;
    }
    // Am realizat conditia 4
    cond4:{
        int num[10]={0};
        cout<<"\tSolutia conditiei 4: \n\t";
        for (i=0;i<strlen(mesaj);i++){
            if (isdigit(mesaj[i])){

```

```

        switch(mesaj[i]){
            case '0': num[0]++; break;
            case '1': num[1]++; break;
            case '2': num[2]++; break;
            case '3': num[3]++; break;
            case '4': num[4]++; break;
            case '5': num[5]++; break;
            case '6': num[6]++; break;
            case '7': num[7]++; break;
            case '8': num[8]++; break;
            case '9': num[9]++; break;
        }
    }
    for (i=0;i<10;i++){
        if(num[i]==0) continue;
        else cout<<"Cifre de "<<i<<" avem "<<num[i]<<".\n";
    }
    return 1;
}
// Am realizat conditia 5
cond5:{
    cout<<"\tSolutia conditiei 5: \n\t";
    cout<<"Determinarea zilei saptamanii conform calendaru-
lui Gregorian.\n";
    ziua=(int) (z+(2.6*1-0.2)-2*(a/100)+a+(a/4)+((a/100)/
4)) % 7;
    cout<<"Eu m-am nascut la "<<z<<". "<<l<<". "<<a<<". ";
    cout<<" Atunci a fost a "<<ziua<<"-a zi din
saptamana."<<endl;
    return 1;
}
}

```

Rezultatele execuției programelor în mediul de programare CodeBlocks:

Rezultatele pentru condiția 1:

Alege o opțiune:

- 1 - Realizarea condiției 1
- 2 - Realizarea condiției 2
- 3 - Realizarea condiției 3
- 4 - Realizarea condiției 4
- 5 - Realizarea condiției 5

Aleg instrucțiunea: 1

Soluția condiției 1:

Bună ziua, părinții mei sunt 65 de ani și 69 de ani. 65 de ani, 24 mai 2009 este ziua mea de naștere.

Rezultatele pentru condiția 2:

Soluția condiției 2:

Bună ziua, părinții mei sunt Alina și Eliza. Azi, 24 mai 2009 asta ziua mea de naștere.

Rezultatele pentru condiția 3:

Soluția condiției 3:

Bună ziua, ai 24 de ani în ziua de azi. Ai, 24 mai 2009 este ziua ta de naștere.

Rezultatele pentru condiția 4:

Soluția condiției 4:

Cifre de 0 avem 2.
Cifre de 2 avem 2.
Cifre de 4 avem 1.
Cifre de 9 avem 1.

Rezultatele pentru condiția 5:

Soluția condiției 4:

Determinarea zilei săptămânii conform calendarului Gregorian.
Eu m-am născut la 24.5.2009.
Atunci a fost a 6-a zi din săptămână.

BIBLIOGRAFIE

- 1 A. Gremalschi, I. Mocanu, I. Spinei, L. Gremalschi, „*Informatică - Manual pentru clasa a 10-a*”, Editura Știința, 2020, 212 pag.
- 2 Мариус Бансила, „*Решение задач на современном C++*”, 2019, 304 стр.
- 3 Алексей Васильев, “*Программирование на C++ в примерах и задачах*”, 2017, 369 стр.
- 4 Михаил Абрамян, „*Введение в стандартную библиотеку шаблонов C++. Описание, примеры использования, учебные задачи*”, 2017, 400 стр.
- 5 Антон Спрол, „*Думай как программист. Креативный подход к созданию кода. C++ версия*”, 2017, 274 стр.
- 6 Tutorial Point, „*Learn C++ programming language*”, 2014, 54 pag.
- 7 Runceanu A., Runceanu M., „*Noțiuni de programare în limbajul C++*”, Editura Academica Brâncuși, 2012, 483 pag.
- 8 Никита Культин, „*C/C++ в задачах и примерах (2-е издание)*”, 2009, 351 стр.
- 9 J. Saulié, „*C++ Language Tutorial*”, 2007, 144 pag.
- 10 Cerchez E., Șerban M., „*Programarea în limbajul C/C++ pentru liceu*”, Editura Polirom, 2005, 296 pag.
- 11 Istrati S. G., „*Inițializare în limbajele C și C++*”, Editura UTM, 2003, 106 pag.
- 12 Ulla Kirch-Prinz, P. Prinz, „*A Complete Guide to Programming in C++*”, 2002, 846 pag.
- 13 Negrescu L., „*Limbajele C/C++ pentru începători*”, Editura Albastră, 1995, 783 pag.
- 14 Steve Oualline, „*Practical C++ Programming*”, 1995, 549 pag.

ANEXE

1. OPERATORII ȘI INSTRUCȚIUNILE LIMBAJULUI C/C++ (selecție)

ABREVIERI:

***N** – nivel de prioritate; **Simbol** – token; **Operator/Operație** –denumire; **Ape-lare** – exemple ilustrative de utilizare; **RV** – tipul rezultatului (**cto** – conform cu tipul operanzilor, **ptr** – pointer, **int** – întreg); **LV** – specifică dacă rezultatul poate avea valoare de locație (*l-value*), *Lv* – valabil pentru orice compilator C/C++, *L** – valabil în C++; **SE** – specifică dacă operatorul produce efecte secundare (*side effects*).*

N	Simbol	Operator / Operație	Apelare	RV	LV	SE
1	::	op. de rezoluție	std::cout	cto	Lv	-
OPERATORI POSTFIXAȚI						
2	()	apel de funcție	funcție(lista)	cto	-	SE
	[]	indexare tablou	tablou[indice]	cto	Lv	-
	. și ->	selecție membru	clasă.membru	cto	Lv	-
	++ și --	incrementare/decrementare	c++ / c--	cto	-	SE
OPERATORI UNARI						
3	++ și --	incrementare/decrementare	++c / --c	cto	L*	SE
	&	op. adresă	&l-valoare	ptr	-	-
	*	op. țintă	*pointer	cto	Lv	-
	+	păstrare semn	+l-valoare	cto	-	-
	-	schimbare semn	-l-valoare	cto	-	-
	~	negare pe biți	~expresie	int	-	-
	!	negare logică	!expresie	bool	-	-
	sizeof	aflarea spațiului de stocare	sizeof(tip)	int	-	-
	(tip)	op. de conversie explicită	(tip)expresie	tip	-	-

N	Simbol	Operator / Operație	Apelare	RV	LV	SE
OPERATORI BINARI						
4	* și ->*	selecție indirectă membru	clasă.*ptr	cto	Lv	-
	* și %	op. multiplicativi	expr * expr	cto	-	-
	+ și -	op. aditivi	expr + expr	cto	-	-
	<< / >>	op. de deplasare pe biți	Expr << deplasare	int	-	-
	<= și >=	op. de comparație	expr < expr expr > expr	bool	-	-
	== !=	op. de egalitate	expr == expr	bool	-	-
	&	op. și pe biți	expr & expr	int	-	-
	^	op. sau exclusiv pe biți	expr ^ expr	int	-	-
		op. sau pe biți	expr expr	int	-	-
	&&	op. și logic	expr && expr	bool	-	-
		op. sau logic	expr expr	bool	-	-
OPERATORI TERNARI						
5	? :	operatorul condițional	cond?Da : Nu	cto	Lv	-
OPERATORI DE ATRIBUIRE						
6	=	atribuire simplă	l-val = expresie	cto	L*	SE
	+= și /- = / și *= /=	atribuire compusă aritmetică	l-val += termen	cto	L*	SE
	/=		l-val *= factor			
	<<=	atribuire compusă pe biți	l-val<< = depl	cto	L*	SE
	&= ^= și =		l-val & = masca			

Nr.	Instrucțiune	Sintaxă și comentarii
INSTRUCȚIUNI SECVENȚIALE		
1	- instrucțiunea declarație	declaratie;
2	- instrucțiunea expresie	expresie; // instrucțiunea nulă: ;
3	- instrucțiunea compusă	{ instructiune instructiune ... instructiune} // blocul nul: {}
INSTRUCȚIUNI DE SELECȚIE (SAU DE DECIZIE)		
4	- if	if (expr-cond) instructiune
5	- if else	if (expr-cond) instructiune else instructiune
6	- switch	switch (expr-selector) instructiune // case ex-eticheta: // default:
INSTRUCȚIUNI CICLICE		
7	- while	while (expr-test) instructiune
8	- do - while	do instructiune while (expr-test);
9	- for	for (expr-init; expr-test; expr-reluare) in- structiune for (; ;) instructiune // ciclare infinită
INSTRUCȚIUNI DE SALT		
10	- break	break; // salt afară din ciclare/selecție
11	- continue	continue; // salt la reluarea ciclării
12	- return	return expr-rezultat; // ieșire din funcție
13	- goto	goto identif-eticheta; // id_eticheta: instructiune

2. FUNCȚII DIN FIȘIERE ANTET (selecție)

Biblioteca IOSTREAM

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>cin (wcin)</i>	<i>citește din fluxul standard de intrare C stdin;</i>
<i>cout (wcout)</i>	<i>scrie în fluxul standard de ieșire C stdout.</i>

Biblioteca IOMANIP

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>resetiosflags</i>	<i>șterge steagurile ios_base specificate;</i>
<i>setiosflags</i>	<i>setează steagurile ios_base specificate;</i>
<i>setbase</i>	<i>modifică baza utilizată pentru I / O întregi;</i>
<i>setfill</i>	<i>schimbă caracterul de umplere;</i>
<i>setprecision</i>	<i>modifică precizia în virgulă mobilă;</i>
<i>setw</i>	<i>modifică lățimea următorului câmp de I / O;</i>
<i>get_money</i>	<i>analizează o valoare monetară;</i>
<i>put_money</i>	<i>formează și realizează o valoare monetară;</i>
<i>get_time</i>	<i>analizează o valoare dată / oră a formatului specificat;</i>
<i>put_time</i>	<i>formează și afișează o dată / oră în funcție de formatul specificat;</i>
<i>quoted</i>	<i>inserează și extrage șiruri citate cu spații încorporate.</i>

Biblioteca CMATH

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>abs</i>	<i>valoarea absolută a unei valori în virgulă mobilă (x);</i>
<i>fmod</i>	<i>restul operației de divizare în virgulă mobilă;</i>
<i>remainder</i>	<i>semnat restul unei operațiuni de divizare;</i>

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>remquo</i>	<i>semnat restul, precum și ultimii trei biți ai operației de divizare;</i>
<i>fma</i>	<i>operație de adăugare multiplă fuzionată;</i>
<i>fmax</i>	<i>mai mare dintre două valori în virgulă mobilă;</i>
<i>fmin</i>	<i>mai mică dintre două valori în virgulă mobilă;</i>
<i>fdim</i>	<i>diferență pozitivă de două valori în virgulă mobilă, adică $\max(0, x-y)$;</i>
<i>nan</i>	<i>nu-un-număr (NaN);</i>
<i>exp</i>	<i>întoarce e ridicat la puterea dată, adică: e^x</i>
<i>exp2</i>	<i>returnează 2 ridicat la puterea dată, adică: 2^x</i>
<i>expm1</i>	<i>întoarce e ridicat la puterea dată, minus unu, adică: $e^x - 1$</i>
<i>log</i>	<i>calculează logaritmul natural (baza e), adică: $\ln(x)$</i>
<i>log10</i>	<i>calculează logaritmul comun (baza 10), adică: $\lg(x)$</i>
<i>log2</i>	<i>calculează logaritmul numărului dat în baza 2, adică: $\log_2 x$</i>
<i>log1p</i>	<i>calculează $\ln(1+x)$</i>
<i>pow</i>	<i>calculează x^y</i>
<i>sqrt</i>	<i>calculează $\sqrt{x}$</i>
<i>cbrt</i>	<i>calculează $\sqrt[3]{x}$</i>
<i>hypot</i>	<i>calculează $\sqrt{x^2+y^2}$ sau $\sqrt{x^2+y^2+z^2}$</i>
<i>sin</i>	<i>calculează $\sin(x)$</i>
<i>cos</i>	<i>calculează $\cos(x)$</i>
<i>tan</i>	<i>calculează tangenta: $\tan(x)$</i>
<i>asin</i>	<i>calculează $\arcsin(x)$</i>

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>acos</i>	<i>calculează $\arcsin(x)$</i>
<i>atan</i>	<i>calculează $\arctan(x)$</i>
<i>atan2</i>	<i>calculează arc tangenta, folosind semne pentru determinarea cadranelor;</i>
<i>ceil</i>	<i>cel mai apropiat număr întreg nu mai mic decât valoarea dată;</i>
<i>floor</i>	<i>cel mai apropiat număr întreg nu mai mare decât valoarea dată;</i>
<i>trunc</i>	<i>cel mai apropiat număr întreg nu mai mare ca mărime decât valoarea dată;</i>
<i>round</i>	<i>cel mai apropiat număr întreg, rotunjind de la zero în cazuri la jumătate;</i>
<i>nearbyint</i>	<i>cel mai apropiat număr întreg folosind modul curent de rotunjire;</i>
<i>rint</i>	<i>cel mai apropiat număr întreg folosind modul curent de rotunjire cu excepție dacă rezultatul diferă;</i>
<i>fpclassify</i>	<i>clasifică valoarea în virgulă mobilă dată;</i>
<i>isfinite</i>	<i>verifică dacă numărul dat are valoare finită;</i>
<i>isinf</i>	<i>verifică dacă numărul dat este infinit;</i>
<i>isnan</i>	<i>verifică dacă numărul dat este NaN;</i>
<i>isnormal</i>	<i>verifică dacă numărul dat este normal;</i>
<i>signbit</i>	<i>verifică dacă numărul dat este negativ;</i>
<i>isgreater</i>	<i>verifică dacă primul argument în virgulă mobilă este mai mare decât al doilea;</i>
<i>isgreaterequal</i>	<i>verifică dacă primul argument în virgulă mobilă este mai mare decât al doilea;</i>
<i>isless</i>	<i>verifică dacă primul argument în virgulă mobilă este mai mic decât al doilea;</i>
<i>islessequal</i>	<i>verifică dacă primul argument în virgulă mobilă este mai mic</i>

Obiectul / Funcția	Funcționalitatea (realizarea)
	<i>sau egal decât al doilea;</i>
<i>islessgreater</i>	<i>verifică dacă primul argument în virgulă mobilă este mai mic sau mai mare decât al doilea;</i>
<i>isunordered</i>	<i>verifică dacă două valori în virgulă mobilă sunt neordonate;</i>

Biblioteca NUMERIC

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>iota</i>	<i>umple un interval cu creșteri succesive ale valorii inițiale;</i>
<i>accumulate</i>	<i>rezumă (sumează) o serie de elemente;</i>
<i>reduce</i>	<i>similar cu std :: accumulate, cu excepția cazului în care o serie de elemente nu este în ordine;</i>
<i>transform_reduce</i>	<i>aplică un invocabil, apoi reduce din funcționare;</i>
<i>inner_product</i>	<i>calculează produsul interior a două game de elemente;</i>
<i>adjacent_difference</i>	<i>calculează diferențele dintre elementele adiacente dintr-un interval;</i>
<i>partial_sum</i>	<i>calculează suma parțială a unei game de elemente;</i>
<i>inclusive_scan</i>	<i>similar cu std :: partial_sum (găsiți deosebirea);</i>
<i>exclusive_scan</i>	<i>similar cu std :: partial_sum (găsiți deosebirea);</i>
<i>gcd</i>	<i>șablonul funcției constexpr returnează cel mai mare divizor comun al a două numere întregi;</i>
<i>lcm</i>	<i>șablonul funcției constexpr returnând cel mai mic multiplu comun din două numere întregi;</i>
<i>midpoint</i>	<i>punctul mediu între două numere sau pointeri.</i>

Biblioteca STRING

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>stoi</i>	convertește un șir într-un număr întreg cu semn;
<i>stoul</i>	convertește un șir într-un număr întreg fără semn;
<i>stof</i>	convertește un șir într-o valoare în virgulă mobilă;
<i>to_string</i>	convertește o valoare integrală sau în virgulă mobilă în șir;
<i>to_wstring</i>	convertește o valoare integrală sau în virgulă mobilă în wstring;
<i>begin</i>	returnează un iterator la începutul unui container sau vector (matrice);
<i>end</i>	returnează un iterator la capătul unui container sau vector (matrice);
<i>rbegin</i>	returnează un iterator invers într-un container sau vector (matrice);
<i>rend</i>	returnează un iterator de capăt invers pentru un container sau vector (matrice);
<i>size</i>	returnează dimensiunea unui container sau vector (matrice);
<i>empty</i>	verifică dacă containerul este gol;
<i>data</i>	obține indicatorul către vectorul (matricea) de bază.

Biblioteca CCTYPE

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>isalnum</i>	Verificați dacă caracterul este alfanumeric;
<i>isalpha</i>	Verificați dacă caracterul este alfabetic;
<i>isblank</i>	Verificați dacă caracterul este gol;
<i>iscntrl</i>	Verificați dacă caracterul este un caracter de control;
<i>isdigit</i>	Verificați dacă caracterul este o cifră zecimală;
<i>isgraph</i>	Verificați dacă caracterul are reprezentare grafică;

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>islower</i>	Verificați dacă caracterul este literă mici;
<i>isprint</i>	Verificați dacă caracterul este imprimabil;
<i>ispunct</i>	Verificați dacă caracterul este un caracter de punctuație;
<i>isspace</i>	Verificați dacă caracterul este un spațiu alb;
<i>isupper</i>	Verificați dacă caracterul este literă mare;
<i>isxdigit</i>	Verificați dacă caracterul este o cifră hexazecimală;
<i>tolower</i>	Conversia literelor mari în minuscule;
<i>toupper</i>	Converțiți litere mici în majuscule.

Biblioteca CSTRING

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>strcpy</i>	copiază un șir la altul;
<i>strncpy</i>	copiază o anumită cantitate de caractere dintr-un șir în altul;
<i>strcat</i>	concatenează două șiruri;
<i>strncat</i>	concatenează o anumită cantitate de caractere din două șiruri;
<i>strxfrm</i>	transformă un șir astfel încât strcmp să producă același rezultat ca strcmp;
<i>strlen</i>	returnează lungimea unui șir dat;
<i>strcmp</i>	compară două șiruri;
<i>strncmp</i>	compară un anumit număr de caractere din două șiruri;
<i>strcoll</i>	compară două șiruri în conformitate cu localizarea curentă;
<i>strchr</i>	găsește prima apariție a unui personaj;
<i>strrchr</i>	găsește ultima apariție a unui personaj;
<i>strspn</i>	returnează lungimea segmentului inițial maxim care constă numai ale caracterelor găsite într-un alt șir de octeți;

Obiectul / Funcția	Funcționalitatea (realizarea)
<i>strcspn</i>	<i>returnează lungimea segmentului inițial maxim care constă numai ale caracterelor care nu se găsesc într-un alt șir de octeți;</i>
<i>strpbrk</i>	<i>găsește prima locație a oricărui personaj dintr-un set de separatoare;</i>
<i>strstr</i>	<i>găsește prima apariție a unui șir de caractere;</i>
<i>strtok</i>	<i>găsește următorul simbol într-un șir de octeți;</i>
<i>memchr</i>	<i>caută o matrice pentru prima apariție a unui caracter;</i>
<i>memcmp</i>	<i>compară două tampoane (buffers);</i>
<i>memset</i>	<i>umple un tampon (buffer) cu un caracter;</i>
<i>memcpy</i>	<i>copiază un tampon (buffer) în altul;</i>
<i>memmove</i>	<i>mută un tampon (buffer) la altul;</i>
<i>strerror</i>	<i>returnează o versiune text a unui cod de eroare dat.</i>

3. MODEL DE TESTARE FINALĂ ASISTATĂ DE CALCULATOR

Competențe specifice la unitatea de curs “Programare structurată”:

1. Translarea algoritmilor de decizie în cadrul unei aplicații de console.
2. Translarea algoritmilor ciclici în cadrul unei aplicații de console.
3. Elaborarea algoritmilor de rezolvare a problemelor pentru tipuri de date simple.
4. Elaborarea algoritmilor de rezolvare a problemelor pentru tipuri de date tablou.

Unități de competență la unitatea de curs “Programare structurată”:

1. Editarea fragmentelor de program.
2. Elaborarea de programe elementare.
3. Scrierea programelor în mediul de programare.
4. Utilizarea instrucțiunilor de decizie în cadrul aplicațiilor de consolă.
5. Utilizarea ciclurilor în cadrul aplicațiilor de consolă.
6. Prelucrarea tablourilor în cadrul aplicațiilor de consolă.
7. Prelucrarea șirurilor de caractere în cadrul aplicațiilor de consolă.

Model de examen realizat pe platforma Moodle

Testarea finală (examenul nu va depăși 180 min, profesorul stabilește timpul probei teoretice și al celei practice) va fi asistat de calculator, acesta va include două probe: *Proba teoretică (25 min)* și *Proba practică (155 min)*.

Proba teoretică va include un test grilă din minim 15 itemi dintr-o bancă de itemi de minim 60 de itemi propuși spre realizare (din această bancă se itemi se vor selecta cei minim 15 itemi corespunzători fiecărei unități de competență în mod aleator).

Proba practică va include un test grilă (itemi de tip eseu) din minim 5 itemi (probleme de rezolvat în limbajul de programare C++) dintr-o bancă de itemi de minim 20 de itemi propuși spre realizare (itemi se vor selecta în același mod ca la proba teoretică).

Nota finală la această probă de examinare se va calcula astfel: $NT = 0.4 \cdot \text{nota obținută la proba teoretică}$, $NP = 0.6 \cdot \text{nota obținută la proba practică}$, astfel nota finală va fi: $N = NT + NP$.

Exemplu: nota la proba teoretică a fost 8 și la proba practică - 6, atunci nota finală la examen va fi: $0.4 \cdot 8 + 0.6 \cdot 6 = 3.2 + 3.6 = 6.8$ (adică va avea 7).

Banca de itemi (model orientativ)⁵

C = Cunoaștere; **A** = Aplicare (înțelegere funcțională); **I** = Integrare (soluționarea problemelor).

	Unități de competență	Numărul de itemi după nivele cognitive			Total itemi
		C (30%)	A (45%)	I (25%)	
1	<i>Editarea fragmentelor de program.</i>	13 (1)	4 (1)	4 (1)	53
2	<i>Elaborarea de programe elementare.</i>	11 (1)	8 (2)		
3	<i>Scrierea programelor în cadrul mediului de programare</i>	6 (1)	7 (2)		
4	<i>Utilizarea instrucțiunilor de decizie în cadrul aplicațiilor de consolă.</i>	7 (1)	4 (1)	6 (1)	17
5	<i>Utilizarea ciclurilor în cadrul aplicațiilor de consolă.</i>	8 (1)	12 (3)	9 (2)	29
6	<i>Prelucrarea tablourilor în aplicațiile de consolă.</i>	5 (1)	6 (1)	10 (1)	21
7	<i>Prelucrarea șirurilor de caractere în cadrul aplicațiilor de consolă.</i>	8 (1)	4 (1)	8 (1)	20
Total itemi:		58	45	37	140
Total itemi în test:		(7)	(11)	(6)	(24)

Barem de notare pentru proba teoretică⁶

Proba teoretică include cei 18 itemi teoretici conform nivelurilor cognitive notate cu **C** și **I**. Fiecare item din nivelul **C** va fi apreciat cu 1p (punct), iar cele din nivelul **I** - cu 2p (puncte). În total vor fi acumulate: 7+22=29 puncte.

-
- 5 Guțu V., doctor habilitat în pedagogie, profesor universitar, USM, „Evaluarea rezultatelor academice”, link de acces: <http://shorturl.at/DJOW6>
- 6 Anexa 5 din „Ghidul metodologic de elaborare a procedurii de evaluare a calificativelor”, link de acces: <http://shorturl.at/dmqZ7>

ITEMI DE CUNOAȘTERE (*model propus*)

Unitatea de competență 1 (*itemi cu răspuns simplu, A sau F*)

1. Bjarne Stroustrup de la Bell Labs a dezvoltat C++ în anii 1980, ca o serie de îmbunătățiri ale limbajului C.
2. În 1985, numele limbajului a fost schimbat de la C cu clase la C++.
3. Limbajele de programare fac parte din categoria limbajelor artificiale, fiind utilizate în procesul de comunicare om-calculator.
4. Realizarea unui program scris în C++ necesită parcurgerea a patru etape: editare, compilare, link-editare și execuție.
5. Linile precedate de semnul # sunt directive, fiind interpretate de ceea ce se numește predecesor.

Unitatea de competență 2 (*itemi cu răspuns simplu, o variantă corectă*)

- | | |
|---|--|
| <ol style="list-style-type: none">1. Manipulatorii de conversie sunt.
A. ws, endl, ends;
B. setw, setfill, setbase;
C. dec, oct, hex;
D. setioflags, resetioflags, setfill.2. Numărul de zecimale cu care se afișează un număr real poate fi setat cu.
A. setf(); B. fill();
C. precision(); D. width(). | <ol style="list-style-type: none">3. Tipul de date ce are reprezentarea pe 1 octet cu semn este.
A. int; B. char; C. short int;
D. nici un răspuns nu este corect.4. Tipul de date ce are reprezentarea pe 1 octet fără semn este.
A. unsigned char;
B. unsigned int;
C. unsigned short int;
D. nici un răspuns nu este corect. |
|---|--|

Unitatea de competență 3 (*itemi cu răspuns simplu, A sau F*)

1. O instrucțiune simplă este formată din cuvinte cheie, expresii și eventual una sau două instrucțiuni corp.
2. Variabila și expresia utilizată în cadrul unei instrucțiuni de atribuire trebuie să aibă același tip de date sau să fie de tipuri compatibile de date.
3. Funcțiile pentru citirea și afișare datelor se află în biblioteca istream.
4. Pentru afișarea datelor întregi sau reale se pot utiliza funcții din biblioteca iomanip.
5. Funcția setw(nr) = nr - precizează pozițiile pe care va fi afișată o valoare întreagă sau reală. Dacă acest nr este mai mic decât numărul pozițiilor necesare afișării unei date, atunci acesta va afișa 1.

Unitatea de competență 4 (*itemi cu răspuns simplu, A sau F*)

1. Structura alternativă dirijează execuția unei secvențe de instrucțiuni în funcție de valoarea unei condiții plasate în blocul de decizie.
2. Mecanismul de execuție a instrucțiunii if este: se evaluează condiția logică, dacă e falsă condiția logică se execută instrucțiunea1, altfel se execută instrucțiunea2.
3. Folosind combinația "if() ... else", putem stabili comenzi care să fie executate și când condiția instrucțiunii "if()" este FALSE.
4. Instrucțiunea switch analizează o problemă în funcție de cazuri. Dacă nici un caz nu se potrivește cu variabila, atunci se va executa o instrucțiune implicită (default).
5. Valorile din clauzele case a instrucțiunii de decizie switch, nu e neapărat să fie doar constante întregi sau caractere.

Unitatea de competență 5 (*itemi cu răspuns simplu, A sau F*)

1. Instrucțiunea FOR este “cerută” dacă descrierea algoritmului spune: “de la valoarea X la valoarea Y”, “pentru primele X valori”, “de X ori”, etc.
2. Instrucțiunea WHILE se mai numește “repetitivă cu test inițial” sau “condiționată anterior”, pentru ca întâi execută secvența ce trebuie repetată, apoi verifică valoarea condiției.
3. Mecanismul de funcționare a instrucțiunea WHILE: se verifică condiția logică, dacă aceasta condiție este adevărată, atunci se repetă instrucțiunea cât timp condiția e adevărată.
4. Instrucțiunea break poate fi folosită numai în două situații, dacă ea apare fără să fie în corpul unei ciclări sau al unui switch avem o eroare la compilare.
5. În interiorul unei bucle while sau do-while, la întâlnirea instrucțiunii continue are loc un salt direct la re-evaluarea expresiei test, iar în cazul instrucțiunii for se trece mai întâi la evaluarea expresiei de reluare și apoi la evaluarea expresiei test.

Unitatea de competență 6 (*itemi cu răspuns simplu, A sau F*)

1. Tabloul unidimensional este o structură de date căreia i se atribuie un nume. Este format dintr-o colecție de elemente de același tip, dispuse contiguu într-un bloc de memorie.

2. Un element al unui tablou poate fi utilizat ca orice alta variabilă. Adresarea unei componente se face prin indicele ei, trecut între paranteze rotunde.
3. Se pot efectua operații asupra fiecărui element al tabloului, nu asupra întregului tablou.
4. Pentru a adăuga un element într-un tablou unidimensional, tot ceea ce trebuie să facem este să mărim dimensiunea logică N a vectorului și să memorăm noul element în căsuța ce am eliberat-o.
5. În limbajul C++ declararea unei matrici se realizează prin specificarea tipului elementelor din matrice, indentificatorul matricii, urmat apoi de dimensiunea acesteia (numarul de linii și de coloane) între acolade.

Unitatea de competență 7 (itemi cu răspuns simplu, o variantă corectă)

1. Denumirea funcției în C++ ce determină și afișează lungimea unui șir de caractere. A. strlen(); B. width(); C. heigh(); D. sir().	3. Funcția de conversie a unui întreg în șir de caractere este itoa(), pentru utilizarea acesteia trebuie inclus header-ul: A. iomanip; B. stdlib; C. stdio; D. string.
2. Denumirea funcției în C++ ce permite copierea dintr-un șir de caractere sursă, cel mult maxLen caractere la sfârșitul unui alt șir de caractere destinație. A. strlen(); B. strncat(); C. strcpy(); D. strcat().	4. Denumirea funcției în C++ ce permite căutarea primei apariții a unui caracter C într-un șir de caractere S. A. strlen(); B. strcmp(); C. strcpy(); D. strchr().

Răspunsuri	1	2	3	4	5
UC 1	A	F	A	A	F
UC 2	C	C	B	A	-
UC 3	A	A	F	A	F
UC 4	A	F	A	A	F
UC 5	A	F	A	A	A
UC 6	A	F	A	A	F
UC 7	A	B	B	D	-

ITEMI DE APLICARE (*model propus*)

Unitatea de competență 1 (<i>itemi cu răspuns simplu, o variantă corectă</i>)	
1. Fie N este o variabilă de tip întreg ce memorează un număr natural. Stabiliți care din expresiile următoare este egală cu zero, dacă și numai dacă N este un număr nedivizibil cu 3. A. $(2-n \% 3) \% 2$; B. $(1-n \% 3) \% 2$; C. $(1-n \% 3) * (2-n \% 3)$; D. $(1-n \% 3) + (2-n \% 3)$;	3. Variabila n este de tip real. Care dintre următoarele expresii are valoarea TRUE, dacă și numai dacă numărul real memorat în variabila x aparține intervalului (5,8] ? A. $(x < 8) \&\& (x \geq 5)$; B. $(x \leq 8) \parallel (x > 5)$; C. $(x > 8) \parallel (x \leq 5)$; D. $(x \leq 8) \&\& (x > 5)$;
2. Variabilele x și y sunt de tip întreg, x memorând valoarea 4, iar y valoarea 2. Care dintre expresiile C++ de mai jos are valoarea 0? A. $x-y!=0$ B. $x+y>x\%y+1$	4. Care este cea mai mică valoare pe care o poate avea expresia: $n/7 - n \% 7$, dacă variabila N memorează un număr natural cu o singură cifră ? A. 0; B. 1.14; C. -6; D. 1.

Unitatea de competență 2 (<i>itemi cu răspuns simplu, o variantă corectă</i>)	
1. Ce va afișa următoarea secvență de program? A. 5 B. 4	<pre>#include <iostream> using namespace std; int main(){ int x=1; x++; x*=10; --x; cout<<x; }</pre>
2. Ce va afișa următoarea secvență de program? A. 1 B. 0	<pre>#include <iostream> using namespace std; int main(){ int x=7; ++x; x%=5; cout<<(x<=5); }</pre>

3. Ce va afișa următoarea secvență de program? A. 4 C. 3 B. 6 D. 8	<pre>#include <iostream> #include <cmath> using namespace std; int main(){ int x=9; x%=3; cout<<pow(x+2,3); }</pre>
4. Ce va afișa următoarea secvență de program? A. 0.9 C. 1.2 B. 0.2 D. 1.6	<pre>#include <iostream> #include <cmath> using namespace std; int main(){ int x=0,y=5; cout<<exp(x)/y; }</pre>

Unitatea de competență 3 (itemi cu răspuns simplu, o variantă corectă)

1. Ce va afișa următoarea secvență de program? A. 9 0 1 B. 9 0 2 C. 9 0 3 D. 9 0 4	<pre>#include <iostream> using namespace std; int main(){ int g=8,f,p=2; g++; f*=g; p+=f; cout<<g<<" "<<f<<" "<<p; }</pre>
2. Ce va afișa următoarea secvență de program? A. 6 3 B. 6 21 C. 9 21 D. 5 15	<pre>#include <iostream> using namespace std; int main(){ int g=6,p; g--; p=(g*(g+1))/2; cout<<g<<" "<<p; }</pre>
3. Ce va afișa următoarea secvență de program? A. 97 47 \$ B. 97 47 * C. 97 47 ! D. 97 47 ?	<pre>#include <iostream> using namespace std; int main(){ char p='a'; cout<<(int)p<<" "; cout<<p-50<<" "<<(char)(p-64); }</pre>
4. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> #include <cmath></pre>

A. 40	using namespace std;
B. 169	int main(){
C. 30	int g,p=40; g=30-4*p;
D. 196	cout<<pow(--g/10,2);
	}

Unitatea de competență 4 (itemi cu răspuns simplu, o variantă corectă)

1. Ce va afișa următoarea secvență de program?	#include <iostream> using namespace std; int main(){ int a=5,b=8,c=3; if(a>b) if(a>c) cout<<a+2; else cout<<c-2; else if(b>c)cout<<b/2; else cout<<c%2; }
A. 4	
B. 7	
C. 5	
D. 1	
2. Ce va afișa următoarea secvență de program?	#include <iostream> #include <cmath> using namespace std; int main(){ float x=9,f; if(x<=7) f=-6*x+2; else cout<<sqrt(pow(x,3)); }
A. 44	
B. 27	
C. 35	
D. 19	
3. Ce va afișa următoarea secvență de program?	#include <iostream> #include <cmath> using namespace std; int main(){ float x=9.6,f=2.7; if(x>ceil(x)) cout<<abs(x); }
A. 2	C. 4
B. 1	D. eroare.

4. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main(){ int x=8,y=3; switch(x%(y)){ case 1: y*=x; case 2: y+=x; case 3: y-=x; } cout<<y; }</pre>
A. y= 24	C. y= 35
B. y= 84	D. y= 38

Unitatea de competență 5 (itemi cu răspuns simplu, o variantă corectă)	
1. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main() { int i,s=0,p=1,n=5; for(i=1;i<=n;i++){ s+=i;p*=i; } cout<<s<<" "<<p; }</pre>
A. 13 150	
B. 13 120	
C. 15 150	
D. 15 120	
2. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main() { int i,s=0,p=0,n=5; for(i=n;i>=0;i--){ s+=i;p+=s%2; } cout<<s<<" "<<p; }</pre>
A. 7 15	
B. 17 3	
C. 15 4	
D. 3 17	
3. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main() { int i,j=0,n=10; for(i=1;i<=n;i++){ if(n%i==0) j++; } cout<<j; }</pre>
A. 5	C. 2
B. 4	D. 3

4. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main() { int i,E=0,n=4; for(i=1;i<=n;i++){ E=E+i*(i+1); } cout<<E; }</pre>
A. 50	C. 20
B. 40	D. 30

Unitatea de competență 6 (itemi cu răspuns simplu, o variantă corectă)

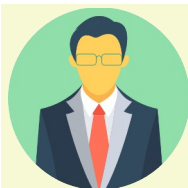
1. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main(){ int n=6,m=5,v1[10]={4,2,1,9,6,8}; int i,j,v2[10]={13,4,9,5,7}; for (i=0;i<n;i++){ for (j=0;j<m;j++){ if(v1[i]==v2[j]) cout<<v1[i]<<" "; } } }</pre>
A. 8 5	
B. 5 8	
C. 4 9	
D. 9 6	
2. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main(){ int i,n=5,s=0,a[10]={1,3,5,2,6}; for (i=0;i<n;++i){ s+=a[i]; cout<<s<<" "; } }</pre>
A. 17 11 9 4 1	
B. 1 17 4 11 9	
C. 9 1 11 17 4	
D. 1 4 9 11 17	
3. Ce va afișa următoarea secvență de program?	<pre>#include <iostream> using namespace std; int main(){ int i=0,n=4,p=1,v[10]={1,3,5,8}; while (i<n-1 && p!=0){ if (v[i]%2==0 && v[i+1]%2==0){ p=0; i++; } } }</pre>
A. Eroare	
B. Nu	
C. Da	
D. Nu afișează nimic	

	<pre> if (p=0) cout<<"Nu"; else cout<<"Da"; } } } </pre>
4. Ce va afișa următoarea secvență de program? A. 0 B. 2 C. 3 D. 5	<pre> #include <iostream> using namespace std; int main(){ int i,n=5,p=0,v[10]={3,-4,5,2,0}; for(i=1;i<n;i++){ if(v[0]>v[i]) p++; } cout<<p; } </pre>

Unitatea de competență 7 (itemi cu răspuns simplu, o variantă corectă)

1. Ce va afișa următoarea secvență de program? A. 3 prfsr B. 5 prfsr C. 3 oeo D. 5 oeo	<pre> #include <iostream> #include <cstring> using namespace std; int main(){ char a[20]="profesor"; int i,k=0; char v[]="aeiouAEIOU"; for(i=0;i<strlen(a);i++){ if(strchr(v,a[i])!=0){ k++; strcpy(a+i,a+i+1); } } cout<<k<<" "<<a; } </pre>
2. Ce va afișa următoarea secvență de program? A. Eroare de program B. InfOrmAtIcA C. INFORMATICA D. iNfOrMaTiCa	<pre> #include <iostream> #include <cstring> using namespace std; int main(){ char s[20]="informatica"; int i; char v[]="aeiouAEIOU"; for(i=0;i<strlen(s);i++){ if(strchr(v,s[i])!=0) s[i]=s[i]+'A'-'a'; cout<<s; } } </pre>
3. Ce va afișa următoarea secvență de program?	<pre> #include <iostream> #include <cstring> </pre>

A. nina B. lina C. alina D. li	<pre>using namespace std; int main(){ char s[20]="annaalina"; int i=0; while(i<strlen(s)-1){ if(s[i]==s[i+1]){ strcpy(s+i,s+i+2); if (i>0)i--; } else i++; } cout<<s; }</pre>
4. Ce va afișa următoarea secvență de program? A. elnoov B. eeoooool C. olonel D. elonov	<pre>#include <iostream> #include <cstring> using namespace std; int main(){ char s[20]="lenovo",aux; int i=0; do{ aux=s[i]; s[i]=s[i+1]; s[i+1]=aux; i=i+2; } while(i<strlen(s)-1); cout<<s; }</pre>



- ✓ *Itemii sugerați ca model pentru aceste două categorii pot fi modificați de fiecare cadru didactic individual.*
- ✓ *Fiecare cadru didactic poate elabora un număr individual de itemi compatibili pentru categoriile de itemi menționate recent.*
- ✓ *Banca de itemi trebuie să fie cât mai voluminoasă!*

ITEMI DE INTEGRARE (*model propus*)

CATEGORIA 1 – Implementarea formulelor matematice

Elaborați un cod C/C++ care va afișa la ecran rezultatul expresiei matematice. În acest exemplu litera A, reprezintă numărul de ordine al elevului în registrul grupei. Afișați soluția pentru funcțiile de mai jos:

$$\begin{array}{ll} \text{a) } f(A) = 2 \cdot \cos\left(\frac{3\pi}{A}\right) \sin\left(\frac{5\pi}{A}\right) - 1 & \text{d) } f(A) = \frac{\sqrt{A}}{e^{A+1}} + \sqrt[5]{9-A} - A^7 \\ \text{b) } g(A) = \frac{\ln(A)}{e^{A+1}} - \frac{\sqrt[3]{A+5}}{|1-\sqrt{A}|} + 2 \cdot A^3 & \text{e) } g(A) = \left(\frac{\sqrt[3]{-A}}{e^A} + A^3\right)^{\frac{3}{2}} - 1 \\ \text{c) } h(A) = 2 \cdot \operatorname{acos}\left(\frac{2\pi}{A}\right) \sqrt{|1-A|} \cdot e^2 & \text{f) } g(A) = \frac{\ln(A)}{e^{1-x}} - \sqrt[4]{2020} \cdot \log(A) \end{array}$$

CATEGORIA 2 – Implementarea instrucțiunilor de decizie

1. Elaborați un cod C/C++ care va determina dacă o valoare X aparține intervalului $[a, b]$. Toate variabilele se vor citi de la tastatură, iar rezultatul se va afișa sub forma „Da” sau „Nu”.
2. Elaborați un cod C/C++ care va determina de câte sticle a câte A litri fiecare este nevoie pentru a umple un vas de B litri. Toate variabilele se vor citi de la tastatură.
3. Elaborați un cod C/C++ care va determina dacă două numere A și B sunt pare și nepozitive. Toate variabilele se vor citi de la tastatură, iar rezultatul se va afișa sub forma „Da” sau „Nu”.
4. Elaborați un cod C/C++ care va determina dacă două numere A și B sunt divizibile cu 3 și 5. Toate variabilele se vor citi de la tastatură, iar rezultatul se va afișa sub forma „Da” sau „Nu”.
5. Elaborați un cod C/C++ care va determina dacă două numere A și B în sumă formează un număr care se împarte la 5. Toate variabilele se vor citi de la tastatură, iar rezultatul se va afișa sub forma „Da” sau „Nu”.
6. Elaborați un cod C/C++ care va determina dacă două numere A și B înmulțite formează un număr care se împarte la 3. Toate variabilele se vor citi de la tastatură, iar rezultatul se va afișa sub forma „Da” sau „Nu”.

CATEGORIA 3 – Implementarea instrucțiunilor ciclice și de salt (opțional)

1. *Elaborați un cod C/C++ care va afișa toate numerele pare divizibile cu 3 din intervalul $[0, N]$. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică FOR.*
2. *Elaborați un cod C/C++ care va afișa toate numerele impare din intervalul $[0, N]$ și suma acestora. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică FOR.*
3. *Elaborați un cod C/C++ care va afișa suma și media aritmetică a tuturor numerelor din intervalul $[0, N]$. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică FOR.*
4. *Elaborați un cod C/C++ care va afișa toate numerele pare divizibile cu 3 din intervalul $[0, N]$. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică WHILE.*
5. *Elaborați un cod C/C++ care va afișa toate numerele impare din intervalul $[0, N]$ și suma acestora. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică WHILE.*
6. *Elaborați un cod C/C++ care va afișa suma și media aritmetică a tuturor numerelor din intervalul $[0, N]$. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică WHILE.*
7. *Elaborați un cod C/C++ care va afișa toate numerele pare divizibile cu 3 din intervalul $[0, N]$. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică DO ... WHILE.*
8. *Elaborați un cod C/C++ care va afișa toate numerele impare din intervalul $[0, N]$ și suma acestora. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică DO ... WHILE.*
9. *Elaborați un cod C/C++ care va afișa suma și media aritmetică a tuturor numerelor din intervalul $[0, N]$. Numărul N este natural, introdus de la tastatură. Se va utiliza instrucțiunea ciclică DO ... WHILE.*

CATEGORIA 4 – Implementarea array-urilor

1. *Elaborați un cod C/C++ care va afișa suma și media aritmetică a numerelor pare nepozitive din tabloul unidimensional A, ce conține N elemente. Numărul N este întreg, introdus de la tastatură.*

2. *Elaborați un cod C/C++ care va afișa suma și media aritmetică a numerelor impare pozitive din tabloul unidimensional A, ce conține N elemente. Numărul N este întreg, introdus de la tastatură.*
3. *Elaborați un cod C/C++ care va afișa elementul maxim și poziția acestuia din tabloul unidimensional A, ce conține N elemente. Numărul N este întreg, introdus de la tastatură.*
4. *Elaborați un cod C/C++ care va afișa elementul minim și poziția acestuia din tabloul unidimensional A, ce conține N elemente. Numărul N este întreg, introdus de la tastatură.*
5. *Elaborați un cod C/C++ care va afișa elementele situate pe diagonala principală a unui tabloul bidimensional A, ce conține NxN elemente întregi. Toate datele vor fi introduse de la tastatură.*
6. *Elaborați un cod C/C++ care va afișa elementele situate pe diagonala secundară a unui tabloul bidimensional A, ce conține NxN elemente întregi. Toate datele vor fi introduse de la tastatură.*
7. *Elaborați un cod C/C++ care va afișa elementele situate deasupra diagonalei secundare a unui tabloul bidimensional A, ce conține NxN elemente întregi. Toate datele vor fi introduse de la tastatură.*
8. *Elaborați un cod C/C++ care va afișa elementele situate deasupra diagonalei principale a unui tabloul bidimensional A, ce conține NxN elemente întregi. Toate datele vor fi introduse de la tastatură.*
9. *Elaborați un cod C/C++ care va afișa elementele situate sub diagonala secundară a unui tabloul bidimensional A, ce conține NxN elemente întregi. Toate datele vor fi introduse de la tastatură.*
10. *Elaborați un cod C/C++ care va afișa elementele situate sub diagonala principală a unui tabloul bidimensional A, ce conține NxN elemente întregi. Toate datele vor fi introduse de la tastatură.*

CATEGORIA 5 – Implementarea string-urilor
--

1. *Elaborați un cod C/C++ care va afișa lungimea unui șir de caractere S. Șirul de caractere va fi citit de la tastatură. Pentru rezolvarea acestei probleme nu se va utiliza funcția `strlen()`.*
2. *Elaborați un cod C/C++ care va afișa un șir de caractere S cu litere majuscule. Șirul de caractere va fi citit de la tastatură. Pentru rezolvarea acestei probleme nu se va utiliza funcția predefinită, ci codul ASCII.*

3. *Elaborați un cod C/C++ care va afișa un șir de caractere S cu litere minuscule. Șirul de caractere va fi citit de la tastatură. Pentru rezolvarea acestei probleme nu se va utiliza funcția predefinită, ci codul ASCII.*
4. *Elaborați un cod C/C++ care va afișa numărul de vocale dintr-un șir de caractere S. Șirul de caractere va fi citit de la tastatură. Pentru determinarea lungimii șirului de caractere se va utiliza funcția strlen().*
5. *Elaborați un cod C/C++ care va afișa numărul de consoane dintr-un șir de caractere S. Șirul de caractere va fi citit de la tastatură. Pentru determinarea lungimii șirului de caractere se va utiliza funcția strlen().*
6. *Elaborați un cod C/C++ care va afișa numărul de cifre dintr-un șir de caractere S. Șirul de caractere va fi citit de la tastatură. Pentru determinarea lungimii șirului de caractere se va utiliza funcția strlen().*
7. *Se citește un text de la tastatură. Elaborați un cod C/C++ care va afișa frecvențele de apariții ale tuturor vocalelor din textul dat. Pentru determinarea lungimii șirului de caractere se va utiliza funcția strlen().*
8. *Se citește un text de la tastatură. Elaborați un cod C/C++ care va afișa frecvențele de apariții ale tuturor consoanelor din textul dat. Pentru determinarea lungimii șirului de caractere se va utiliza funcția strlen().*



- ✓ *Itemii recomandați ca model pentru această ultimă categorie pot fi modificați sau supliți de fiecare cadru didactic în mod individual.*
- ✓ *Fiecare elev poate studia acești itemi model pentru a exersa sau a verifica cunoștințele proprii dobândite la unitatea de curs respectivă.*
- ✓ *Profesorii să elaboreze mai mulți itemi în banca de itemi de pe platforma (Moodle) pentru testare finală a capacităților individuale ale elevilor.*

Barem de corectare și notare pentru proba practică

Nr.	Motivarea punctajului acordat	Puncte	Total
1	- Definirea și declararea corectă a datelor. - Introducerea corectă a datelor de intrare. - Aplicarea corectă a operatorilor și funcțiilor corespunzătoare. - Afișarea răspunsului corect la consolă.	1 1 2 1	6
2	- Definirea și declararea corectă a datelor. - Introducerea corectă a datelor de intrare. - Aplicarea corectă a operatorilor. - Aplicarea corectă a sintaxei instrucțiunii de decizie corespunzătoare - Afișarea răspunsului corect la consolă.	1 1 2 2 1	7
3	- Definirea și declararea corectă a datelor. - Introducerea corectă a datelor de intrare. - Aplicarea corectă a operatorilor. - Aplicarea corectă a sintaxei instrucțiunii de decizie - Aplicarea corectă a sintaxei instrucțiunii de ciclare - Corectitudinea algoritmului - Afișarea răspunsului corect la consolă.	1*2 1*2 2*2 2*2 2*2 1*2 1*2	20
4	- Definirea și declararea corectă a datelor. - Introducerea corectă a datelor de intrare. - Aplicarea corectă a operatorilor. - Aplicarea corectă a sintaxei instrucțiunii de decizie corespunzătoare. - Aplicarea corectă a sintaxei instrucțiunii de ciclare - Aplicarea corectă a tipului de date tablou - Corectitudinea algoritmului - Afișarea răspunsului corect la consolă.	1 1 1 1 1 2 1 1	9
5	- Definirea și declararea corectă a datelor. - Introducerea corectă a datelor de intrare. - Aplicarea corectă a sintaxei instrucțiunii de decizie corespunzătoare. - Aplicarea corectă a sintaxei instrucțiunii de ciclare	1 1 1 1	7

Nr.	Motivarea punctajului acordat	Puncte	Total
	• <i>Aplicarea corectă a tipului de date string</i> • <i>Corectitudinea algoritmului</i> • <i>Afișarea răspunsului corect la consolă.</i>	1 1 1	
Punctaj total acumulat:		49	

Nota	10	9	8	7	6	5	4	3	2	1
Punctaj	49-47	46-43	42-38	37-31	30-24	23-16	15-10	9-5	4-3	2-0

NOTIȚE

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.