**1.** Dintre expresiile C/C++ de mai jos, cea care are valoarea 1 dacă și numai dacă numărul memorat în variabila întreagă x aparține intervalului închis [-5, 5]:

a. (x>-5) && (x<5)

b. abs(x) <= 5

c. !(x<-5) || !(x>5)

d. !((x<-5) && (x>5))

**2.** Utilizând metoda backtracking, se generează toate valorile naturale formate cu cifre ale numărului 247 și care sunt strict mai mici ca acesta. Primele șase numere generate sunt, în această ordine, 2, 22, 222, 224, 227, 24. Indicați al câtelea număr generat este 7.

a. al 11-lea

b. al 12-lea

c. al 13-lea

d. al 14-lea

**3.** Se consideră variabila t, declarată mai jos, care memorează coordonatele în planul xOy ale vârfurilor A, B și C ale unui triunghi:

struct punct {

int x,y;

};

struct triunghi {

struct punct A, B, C;

};

struct triunghi t;

Care dintre următoarele instrucțiuni inițializează cu valoarea 1 coordonata y a vârfului B al triunghiului respectiv?

a. t.B[y] = 1;

b. t->B->y = 1;

c. t.B.y = 1;

d. punct(t,B,y) = 1;

**4.** În secvența de program de mai jos variabila a memorează un tablou bidimensional cu 10 linii și 10 coloane, numerotate de la 1 la 10, cu elemente numere întregi.

for (int i=1; i<=10; i++)

for (int j=1; j<=10; j++)

a[i][j] = (i%2)\*i + (j%2)\*j;

Valoarea sumei elementelor de pe diagonala secundară a tabloului construit în urma executării acestei secvențe este:

a. 48

b. 49

c. 50

d. 60

**5.** Un graf orientat are următoarele arce: (4,2), (2,3), (3,5), (2,1), (5,3), (1,2)

Numărul minim de arce care trebuie adăugate acestui graf pentru ca el să devină tare conex este:

a. 1

b. 2

c. 3

d. 0 (graful dat este tare conex)

**6.** Algoritmul de mai jos este reprezentat în pseudocod.

citeste n,k (numere naturale nenule, fiecare avand cel mult 3 cifre)

t <- 0

┌cât timp n≥1 execută

│┌dacă n>k atunci i <- k

││altfel i <- n

│└■

│ t <- t+1

│ n <- n-i

│┌cât timp i≥1 execută

││ scrie t,' '

││ i <- i-1

│└■

└■

Observație: scrie t,' ' are semnificația că după valoarea lui t se afișează exact UN spațiu.

**6a.** Scrieți numerele care se afișează în urma executării algoritmului, în ordine, dacă pentru n se citește valoarea 7, iar pentru k se citește valoarea 3.

(6 puncte, 3 submit-uri)

**6b.** Dacă pentru variabila k se citește valoarea 7, scrieți trei valori distincte care pot fi citite pentru variabila n astfel încât, în urma executării algoritmului, ultimul număr care se afișează, pentru fiecare dintre aceste valori, să fie 7.

(6 puncte, 3 submit-uri)

**6c.** Scrieți programul C/C++ corespunzător algoritmului dat.

(10 puncte, 10 submituri)

**6d.** Scrieți **pe foaia de concurs** în pseudocod un algoritm, echivalent cu cel dat, înlocuind a doua structură cât timp...execută cu o structură repetitivă cu test final.

(6 puncte)

**7.** Subprogramul f este definit astfel:

void f(int n)

{

cout << n - 1 << " ";

if (n > 1)

{

f(n - 2);

cout << n << " ";

}

}

Scrieți în fereastra platformei de concurs ce se afișează în urma apelului f(7).

(6 puncte, 3 submit-uri)

**8.** Un arbore cu nodurile etichetate cu numere naturale începând cu 1 are următorul vector de tați: *13 7 2 13 3 7 0 7 10 13 13 11 6 13*

Indicați vârfurile adiacente cu vârful 13.

(6 puncte, 3 submit-uri)

**9.** Se consideră subprogramul divprimi, cu trei parametri:

*n*, prin care primește o valoare naturală (0 < n < 100000);

*mic* și *mare* de tip int prin care furnizează cel mai mic și respectiv cel mai mare divizor prim pozitiv al lui n sau valoarea 0, prin amândoi, dacă numărul nu are cel puțin doi astfel de divizori.

Scrieți definiția completă a subprogramului. Exemplu: pentru numărul n=20, în urma apelului, mic=2 și mare=5, iar pentru numerele n=9 sau n=23, în urma apelului, mic=0 și mare=0.

ATENȚIE: Scrieți în fereastra de editare definiția funcției cerute. Dacă doriți, puteți defini și alte funcții pe care funcția cerută le apelează, caz în care veți scrie în fereastra platformei și definițiile acestor funcții. Dacă rezolvarea dumneavoastră folosește elemente aflate în diverse biblioteci, altele în afară de iostream veți scrie în fereastra platformei de concurs și declarația de includere pentru acestea. NU scrieți în fereastra de editare funcția main() !

(10 puncte, 10 submit-uri)

**10.** Un text cu cel puțin un caracter și cel mult 100 de caractere este format din cuvinte și numere. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Numerele din text sunt numere naturale cu cel mult 5 cifre. Cuvintele și numerele sunt separate prin exact un spațiu. Textul NU începe și NU se termină cu spații.

Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și afișează pe ecran pe linii separate, cuvintele din text urmate de numerele din text. Cuvintele vor fi afișate în ordinea în care apar în textul citit iar numerele vor fi afișate în ordine inversă față de ordinea în care apar în textul citit.

Dacă textul citit nu conține nici un număr, după cuvintele afișate se va afișa mesajul *fara numere*

Dacă textul citit nu conține nici un cuvânt, se va afișa o primă linie cu mesajul *fara cuvinte*, apoi pe următoarele linii numerele din textul citit în ordine inversă.

Exemple:

dacă se citește textul:

ana are 12 mere si 37 pere

se afișează pe ecran:

ana

are

mere

si

pere

37

12

dacă se citește textul:

alfa beta gamma

se afișează pe ecran:

alfa

beta

gama

fara numere

dacă se citește textul:

20 30 40

se afișează pe ecran:

fara cuvinte

40

30

20

(10 puncte, 10 submit-uri)

**11.** Fișierul bac.in conține un șir de cel puțin trei și cel mult 3000000 de numere întregi, fiecare având cel mult 8 cifre. Numerele sunt separate prin câte un spațiu. Se consideră că un șir format din cel puțin trei termeni formează o progresie aritmetică dacă diferența dintre oricare termen al acestuia și cel aflat pe poziția consecutivă în șir este nenulă și constantă. Rația unei progresii aritmetice este egală cu diferența dintre un termen al acesteia și termenul aflat pe poziția precedentă.

Se cere să se citească numerele din fișierul *bac.in* și să se afișeze pe ecran separate prin exact un spatiu, numărul maxim de termeni ai unei secvențe din șir, secvență care formează o progresie aritmetică precum și rația acestei progresii aritmetice. Dacă există mai multe secvențe care formează progresii aritmetice cu număr maxim de termeni, se va afișa rația cea mai mică a uneia dintre aceste progresii. Dacă nu există nicio astfel de secvență, se afișează pe ecran mesajul *NU EXISTA*. Pentru determinarea numerelor cerute se va utiliza un algoritm eficient din punctul de vedere al timpului de executare și al memoriei folosite.

Exemple: dacă fișierul bac.in conține numerele

2 4 6 8 12 11 10 9 5 1 0 -2 -4 -6

se afișează pe ecran

4 -2

Explicații: șirul din fișier conține următoarele secvențe care formează progresii aritmetice:

2 4 6 8 => număr de termeni = 4, rație = 2

12 11 10 9 => număr de termeni = 4, rație = -1

9 5 1 => număr de termeni = 3, rație = -4

0 -2 -4 -6 => număr de termeni = 4, rație = -2

Dacă fișierul bac.in conține numerele

1 2 4 1 3 7

sau numerele

2 2 2 2 2

se afișează

NU EXISTA

**11a.** Descrieți **pe foaia de concurs** în limbaj natural algoritmul utilizat, justificând eficiența acestuia.

(2 puncte)

**11b.** Scrieți programul C/C++ corespunzător algoritmului descris.

(8 puncte, 10 submituri)