

PROBA SCRISĂ LA INFORMATICĂ
LIMBAJUL C / C++

Filiera teoretică, profilul real, specializările:

- matematică – informatică,
- matematică – informatică intensiv informatică

Filiera vocațională, profilul militar, specializarea matematică – informatică

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.

SUBIECTUL I

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Indicați care dintre expresiile **C/C++** de mai jos are valoarea 1 dacă și numai dacă numărul natural memorat în variabila **x** are cifra unităților egală cu 5 și este un multiplu al numărului natural nenul memorat în variabila **y**. **(4p.)**

a. $x \% 10 == 5 \ \&\& \ x \% y == 0$

b. $x \% 10 == 5 \ || \ x \% y == 0$

c. $x \% 5 == 0 \ \&\& \ x \% y == 0$

d. $x \% 10 == 5 \ \&\& \ y \% x == 0$

2. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu $x \% y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

- a. Scrieți numărul care se afișează în urma executării algoritmului dacă pentru variabila n se citește valoarea 103456, iar pentru variabila m valoarea 1234. **(6p.)**

- b. Scrieți toate perechile de valori naturale distincte care pot fi citite pentru variabilele n și m astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, numărul afișat să fie 10. **(6p.)**

```
citeste n, m
    (numere naturale)
p ← 1
┌cât timp m > 0 execută
| c ← m % 10
| m ← [m / 10]
| n ← n + p * c
| p ← p * 10
└─┘
scrie n
```

- c. Scrieți în pseudocod un algoritm care să nu folosească structuri repetitive și care să fie echivalent cu cel dat. **(4p.)**

- d. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

SUBIECTUL II

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Indicați numărul de circuite distincte ale grafului orientat cu 6 vârfuri, numerotate de la 1 la 6, reprezentat prin listele de adiacență alăturate. Se consideră că un circuit este format numai din arce distincte. Două circuite sunt distincte dacă diferă prin cel puțin un arc. (4p.)
- 1: 3
2: 1, 5, 6
3: 2
4: 3
5: listă vidă
6: 4, 5

a. 0 b. 1 c. 2 d. 3

2. Numim înălțime a unui arbore cu rădăcină numărul de muchii ale celui mai lung lanț elementar care are una dintre extremități în rădăcina arborelui. Înălțimea arborelui cu rădăcină, având 8 noduri, numerotate de la 1 la 8, reprezentat prin vectorul "de tați" (6, 6, 5, 0, 6, 4, 4, 7) este: (4p.)

a. 2 b. 3 c. 4 d. 5

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Fiecare dintre variabilele **A** și **B**, declarate alăturat, memorează coordonatele (**x** abscisa, iar **y** ordonata) câte unui punct în sistemul de coordonate **xOy**. Scrieți o expresie **C/C++** care are valoarea **1** dacă și numai dacă segmentul cu capetele în punctele corespunzătoare variabilelor **A** și **B** intersectează axa **Oy** a sistemului de coordonate. (6p.)
- ```
struct punct
{
int x, y;
} A, B;
```

4. În secvența de program alăturată variabila **a** memorează un șir cu cel mult **100** de caractere, iar variabilele **i** și **k** sunt de tip întreg. Scrieți șirul afișat pe ecran în urma executării secvenței. (6p.)
- ```
k='a'-'A';
strcpy(a,"ExaMeN");
for(i=0; i<strlen(a); i++)
if(a[i]>='a' && a[i]<='z')
a[i]=a[i]-k;
else
a[i]=a[i]+k;
cout<<a; | printf("%s",a);
```

5. Scrieți un program **C/C++** care citește de la tastatură două numere naturale **n** și **m** ($2 < n \leq 24$, $2 < m \leq 24$) și construiește în memorie un tablou bidimensional cu **n** linii și **m** coloane, în care orice element aflat pe prima linie sau pe prima coloană are valoarea **1** și oricare alt element din tablou este egal cu ultima cifră a sumei celor două elemente alăturate lui, aflate pe aceeași linie dar pe coloana din stânga, respectiv pe aceeași coloană, dar pe linia anterioară. Programul afișează pe ecran tabloul obținut, câte o linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

Exemplu: pentru **n=4** și **m=5** se afișează pe ecran tabloul alăturat. (10p.)

```
1 1 1 1 1
1 2 3 4 5
1 3 6 0 5
1 4 0 0 5
```

SUBIECTUL III

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Utilizând metoda backtracking se generează în ordine crescătoare toate numerele naturale de câte patru cifre din mulțimea $A=\{1,2,3,4,5\}$, numere care nu conțin două cifre impare alăturate. Primele opt numere generate sunt, în această ordine, **1212, 1214, 1221, 1222, 1223, 1224, 1225, 1232**. Ultimul număr generat este: **(4p.)**

a. 5455

b. 5454

c. 5452

d. 5445

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Se consideră subprogramul **f**, definit alăturat. Scrieți ce valori au **f(0)**, respective **f(123456)**. **(6p.)**

```
int f (long x)
{ if(x>=123) return 1+f(x/10);
  return 123;
}
```

3. Subprogramul **sub** are trei parametri:

- **n**, prin care primește un număr natural ($0 < n < 100$);
- **v**, prin care primește un tablou unidimensional care memorează un șir de **n** numere naturale, fiecare având cel puțin două cifre și cel mult patru cifre;
- **s**, un număr natural ($0 \leq s < 18$).

Subprogramul determină și afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele din șir care au suma dintre cifra unităților și cifra zecilor strict mai mare decât **s**. Dacă șirul nu conține niciun astfel de număr, subprogramul afișează pe ecran mesajul **NU EXISTA**. În urma apelului, tabloul transmis subprogramului ca parametru nu se modifică.

Scrieți în limbajul **C/C++** definiția completă a subprogramului **sub**.

Exemplu: pentru **n=10**, **v=(351,149, 3794, 502, 785, 258, 258, 1512, 489, 693)**, **s=12** în urma apelului, subprogramul afișează pe ecran, în această ordine, valorile: **3794 785 489 258 258 149**. **(10p.)**

4. Fișierul **BAC.TXT** conține un șir de cel mult **1000** de numere naturale, despărțite prin câte un spațiu, fiecare număr având cel mult **9** cifre. Cel puțin două numere din fișier sunt pătrate perfecte. Se citește șirul din fișier și se cere ca, utilizând un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de executare, să se determine și să se afișeze pe ecran lungimea maximă a unei secvențe a șirului care începe și se încheie cu câte un număr pătrat perfect. O secvență este formată din termeni aflați pe poziții consecutive în șir. Lungimea unei secvențe este egală cu numărul de termeni ai acesteia.

Exemplu: dacă fișierul **BAC.TXT** conține numerele **11 245 36 67 8 576 11 16 41 144 67 89 1011** pe ecran se afișează **8** (corespunzător secvenței **36 67 8 576 11 16 41 144**).

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**

b) Scrieți programul **C/C++** corespunzător algoritmului descris. **(6p.)**