

Testul 7

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei ore.
- Identifierii utilizati în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare muchie are extremități distincte și oricare două muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabilele **x** și **y** sunt de tip întreg. Indicați o pereche de valori pentru care expresia alăturată are valoarea 1.

a. **x=1234** și **y=12** b. **x=1100** și **y=10** c. **x=1003** și **y=3** d. **x=35** și **y=35**

2. Subprogramele **f1** și **f2** sunt definite mai jos. Indicați valoarea **f2(41382)**.

```
int f1(int c)
{
    if (c%2==1) return 1;
    else return 2;
}

int f2(int n)
{
    if (n==0) return 0;
    else return f1(n%10)+f2(n/10);
}
```

- a. 7 b. 8 c. 9 d. 10

3. Utilizând metoda backtracking se generează în ordine alfabetică anagramele cuvântului **bine**. Primele patru soluții generate sunt, în această ordine: **bein**, **beni**, **bien**, **bine**. Indicați a **saptea** soluție generată.

- a. **einb** b. **eibn** c. **ebni** d. **ebin**

4. Un arbore cu rădăcină, cu 7 noduri, numerotate de la 1 la 7, este reprezentat prin vectorul de „tați” (5, 1, 5, 1, 0, 7, 5). Indicați etichetele tuturor nodurilor de tip „frunză”.

- a. 2 3 4 6 b. 1 3 7 c. 2 4 d. 5

5. Un graf neorientat are 5 noduri, etichetate cu câte o literă distinctă din cuvântul **lista**, în care orice nod etichetat cu o vocală este adiacent doar cu nodurile etichetate cu consoane, iar orice nod etichetat cu o consoană este adiacent doar cu nodurile etichetate cu vocale. Indicați numărul de muchii ale acestui graf.

- a. 12 b. 6 c. 4 d. 3

SUBIECTUL al II-lea

(40 de puncte)

1. **Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu **a**%**b** restul împărțirii numărului natural **a** la numărul natural nenul **b** și cu **[c]** partea întreagă a numărului real **c**.

- a. Scrieți valoarea afișată dacă se citesc, în această ordine, numerele 12, 7, 354, 9, 1630, 0. (6p.)

- b. Scrieți o secvență de numere pare care pot fi citite, în această ordine, astfel încât, în urma executării algoritmului, să se afișeze valoarea 751. (6p.)

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

- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adevarat prima structură **cât timp...execută** cu o structură repetitivă cu test final. (6p.)

citește **x** (număr natural)

n←0

cât timp **x**≠0 execută

y←**x**; **c**←0

 cât timp **y**>0 execută

 dacă **y**%10>**c** atunci

c←**y**%10

y← [**y**/10]

n←**n***10+**c**

 citește **x** (număr natural)

 scrie **n**

2. În declarările alăturate, variabila **dataEveniment** memorează data calendaristică la care s-a desfășurat un anumit eveniment, iar variabila **e** memorează numele și data nașterii unui elev.
Scrieți o secvență de instrucțiuni C/C++ în urma executării căreia să se afișeze pe ecran mesajul **ulterior** dacă elevul s-a născut ulterior desfășurării evenimentului sau mesajul **nu** în caz contrar. **(6p.)**
- ```

struct data
{ int zi,luna,an;
}dataEveniment;
struct elev
{ char nume[30];
 data dataNasterii;
}e;

```
3. Variabilele **i** și **j** sunt de tip întreg, iar variabilele **s** și **t** permit memorarea unui sir de cel mult 20 de caractere. Scrieți sirul accesat prin variabila **s** în urma executării secvenței de mai jos.
- ```

strcpy(s,"ABCDUECDA");
i=0; j=strlen(s)-1;
while (i<j)
  if(s[i]==s[j])
    {strcpy(t,s+j+1);strcpy(s+j,t);strcpy(t,s+i+1);strcpy(s+i,t);j=j-2;}
  else { i=i+1; j=j-1; }

```
- (6p.)**

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul **afisare** are trei parametri:
- **x** și **y**, prin care primește câte un număr natural din intervalul $[0, 10^6]$ ($x \leq y$);
 - **k**, prin care primește un număr natural ($k \in [2, 10^2]$).
- Subprogramul afișează pe ecran, în ordine strict crescătoare, numerele din intervalul $[x, y]$, în secvențe de câte **k**, cu excepția ultimei secvențe care poate conține mai puțin de **k** numere. Fiecare secvență se încheie cu câte un simbol *, iar numerele și simbolurile sunt separate prin câte un spațiu, ca în exemplu.
Scrieți definiția completă a subprogramului.
- Exemplu:** dacă **x=11**, **y=21** și **k=4** se afișează pe ecran numerele de mai jos, în acest format.
- ```

11 12 13 14 * 15 16 17 18 * 19 20 21 *

```
- (10p.)**
2. Scrieți un program C/C++ care citește de la tastatură un număr natural, **x** ( $x \in [1, 10^9]$ ), și construiește în memorie un tablou bidimensional, pentru care atât numărul de linii, cât și numărul de coloane sunt egale cu numărul de cifre ale lui **x**, iar elementele fiecărei linii au ca valori cifrele lui **x**, în ordine, ca în exemplu.
- Elementele tabloului obținut sunt afișate pe ecran, linie cu linie, fiecare linie a tabloului pe **1 3 5 9**  
**1 3 5 9**  
**1 3 5 9**  
**1 3 5 9**câte o linie a ecranului, cu elementele de pe aceeași linie separate prin câte un spațiu.
- Exemplu:** dacă **x=1359**, se afișează tabloul alăturat. **(10p.)**
- |          |          |          |          |
|----------|----------|----------|----------|
| <b>1</b> | <b>3</b> | <b>5</b> | <b>9</b> |
|----------|----------|----------|----------|
3. Fișierul **bac.txt** conține cel mult  $10^6$  cifre, separate prin câte un spațiu.  
Se cere să se afișeze pe ecran, separate prin câte un spațiu, toate cifrele pare care apar în fișier sau mesajul **nu există**, dacă nu există astfel de cifre. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
- Exemplu:** dacă fișierul conține cifrele **3 3 0 8 2 1 2 1 3 7 1 5 2 7 1 0 3 2 3** pe ecran se afișează, de exemplu în ordine crescătoare, cifrele **0 0 2 2 2 2 8**
- Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
  - Scrieți programul C/C++corespunzător algoritmului proiectat. **(8p.)**