

Simulare pentru EXAMENUL DE BACALAUREAT - decembrie 2024

Proba E.d)  
INFORMATICĂ

VARIANTA 1

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

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați 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 arc/muchie are extremități distincte și oricare două arce/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. Indicați numerele pe care le pot memora variabilele întregi  $x$  și  $y$ , astfel încât valoarea expresiei `C/C++` alăturate să fie 1.  $x/2+x*y-x/y==0$
- a.  $x=4$  și  $y=2$                                                   b.  $x=6$  și  $y=3$
- c.  $x=8$  și  $y=4$                                                   d.  $x=10$  și  $y=0$
2. Subprogramul `f` este definit alăturat. Indicați ce se afișează în urma apelului de mai jos. `f(75, 30);`
- ```
void f(int i, int j)
{ cout<<i<<" "; | printf("%d ",i);
  if(i!=j)
  { if(i<j){ i=i+j; j=i-j; i=i-j; }
    f(i-j,j);
  }
}
```
- a. 75 30 45                                          b. 75 45 15                                          c. 75 45 15 15                                          d. 75 30 45 15
3. Utilizând metoda backtracking, se generează toate băuturile obținute amestecând sucurile a cel puțin două fructe distincte din mulțimea {**afine**, caise, lămâi, mere, pere}. Primele cinci soluții obținute sunt, în această ordine: (**afine**, caise), (**afine**, caise, lămâi), (**afine**, caise, lămâi, mere), (**afine**, caise, lămâi, mere, pere) și (**afine**, caise, lămâi, pere). A șasea soluție este:
- a. (**afine**, caise, mere)                                                  b. (**afine**, caise, mere, pere)
- c. (**afine**, mere, pere)                                                  d. (**afine**, lămâi, mere, pere)
4. Un graf neorientat cu 8 noduri, numerotate de la 1 la 8, are muchiile [1,2], [1,6], [4,6], [3,6], [6,5], [5,3], [3,4], [7,8], [8,2]. Trei noduri care nu aparțin niciunui ciclu în acest graf pot fi:
- a. 1, 3, 4                                          b. 2, 7, 8                                          c. 3, 5, 6                                          d. 5, 6, 8
5. Un arbore cu 19 noduri, numerotate de la 1 la 19, are ca rădăcină nodul numerotat cu 1; nodul 1 are un singur fiu, și anume nodul 2, iar fiecare nod  $i$  ( $i \geq 2$ ) fie este frunză, fie are drept fii (descendenți direcți) noduri numerotate cu valori din intervalul  $[i+1, 2 \cdot i - 1]$ . Numărul maxim de frunze ale arborelui este:
- a. 9                                                  b. 12                                                  c. 13                                                  d. 16



**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 citește numărul 27102. **(6p.)**
- b) Scrieți trei numere distincte din intervalul  $[10, 9999]$  care pot fi citite, astfel încât în urma executării algoritmului, pentru fiecare dintre acestea, valoarea afișată să fie 11. **(6p.)**
- c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind prima structură **cât timp...execută** cu o structură de tip **pentru...execută**. **(6p.)**

```

citește n
    (număr natural)
m ← 0; x ← 1
cât timp x ≤ 9 execută
    cp ← n
    cât timp cp ≠ 0 execută
        dacă cp % 10 = x atunci
            m ← m * 10 + x
        cp ← [cp / 10]
    x ← x + 1
scrie m
    
```

- 2. Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, având inițial toate elementele nule.

Fără a utiliza alte variabile decât cele menționate, scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila  $a$  să memoreze tabloul alăturat.

```

for (i=1; i<=5; i++)
    for (j=1; j<=5; j++)
        .....
    
```

|   |   |   |   |   |
|---|---|---|---|---|
| 4 | 4 | 4 | 4 | 4 |
| 4 | 3 | 3 | 3 | 3 |
| 4 | 3 | 2 | 2 | 2 |
| 4 | 3 | 2 | 1 | 1 |
| 4 | 3 | 2 | 1 | 0 |

**(6p.)**

- 3. În declarațiile alăturate, variabila  $p$  memorează coordonatele unui punct în sistemul de coordonate  $xOy$ , iar variabila  $c$  memorează datele caracteristice ale unui cerc: lungimea razei și coordonatele centrului său. Scrieți o expresie care are valoarea 1 dacă și numai dacă punctul corespunzător variabilei  $p$  coincide cu centrul cercului corespunzător variabilei  $c$ . **(6p.)**

```

struct punct
{
    int x, y;
} p;
struct cerc
{
    struct punct centru;
    float raza;
} c;
    
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

- 1. Subprogramul **MaxImp** are doi parametri,  $a$  și  $b$ , prin care primește câte un număr natural ( $2 \leq a < b \leq 400$ ). Subprogramul returnează cel mai mare număr natural din intervalul  $[a, b]$  pentru care produsul divizorilor săi impari pozitivi este strict mai mare decât el însuși sau 0, dacă nu există niciun astfel de număr. Scrieți definiția completă a subprogramului.  
**Exemplu:** dacă  $a=14$  și  $b=19$ , atunci subprogramul returnează 18 ( $1 \cdot 3 \cdot 9 = 27 > 18$ ). **(10p.)**
- 2. Într-un text de cel mult 50 de caractere cuvintele sunt separate prin câte un spațiu și sunt formate din litere mari ale alfabetului englez, urmate eventual de caracterul . (punct), dacă sunt scrise prescurtat. Textul reprezintă numele unei instituții de învățământ și doar cuvintele din mulțimea {COLEGIUL, LICEUL, NATIONAL, TEORETIC} pot fi prescurtate, eliminându-se ultimele lor litere. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat și construiește în memorie, apoi afișează pe ecran, numele instituției scris fără prescurtări.  
**Exemplu:** dacă se citește textul COLEG. NATIONAL DE INFORMATICA sau textul COLEG. NAT. DE INFORMATICA se obține COLEGIUL NATIONAL DE INFORMATICA **(10p.)**
- 3. Șirul  $f$  este definit astfel:  $f_1=x$ ;  $f_2=y$ ;  $f_3=z$ ;  $f_i=f_{i-1}+f_{i-2}-f_{i-3}$ , unde  $x, y, z$  și  $i$  sunt numere naturale nenule,  $i > 3$ .  
De exemplu, dacă  $x=1$ ,  $y=2$  și  $z=4$  șirul este: 1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, ...  
Se citesc de la tastatură un număr natural,  $n$  ( $n \in [1, 10^4]$ ), apoi trei numere naturale din intervalul  $[1, 10^2]$ ,  $x, y$  și  $z$ , reprezentând, în această ordine, primii trei termeni ai șirului precizat mai sus. Se cere să se scrie în fișierul **bac.txt** primii  $n$  termeni ai șirului, separați prin câte un spațiu, în ordine inversă a apariției lor în șir. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.  
**Exemplu:** dacă  $n=10$ ,  $x=1$ ,  $y=2$  și  $z=4$  fișierul conține numerele: 14 13 11 10 8 7 5 4 2 1  
a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**  
b) Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**