

Examenul de bacalaureat național 2020  
Proba E. d)  
Informatică  
Limbajul C/C++

Testul 15

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică  
Filieră vocațională, profil militar, specializare matematică-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. Expresia C/C++  
 $(x \geq 16) \ \&\& \ ! ( \ x < 17 \ || \ x > 19 ) \ \&\& \ (x \leq 20)$   
are valoarea 1 dacă și numai dacă valoarea memorată de variabila întreagă  $x$  aparține intervalului:
- a. [16,18]                      b. [17,19]                      c. [18,20]                      d. [19,20]
2. Utilizând metoda backtracking se generează toate posibilitățile de a așeza în compartimentele unei voliere porumbei de rase din mulțimea {creți, iacobini, jucători, rotați, toboșari}. Două soluții sunt diferite dacă ordinea vaselor diferă. Primele patru soluții obținute sunt, în această ordine: (creți, iacobini, jucători, rotați, toboșari), (creți, iacobini, jucători, toboșari, rotați), (creți, iacobini, rotați, jucători, toboșari), (creți, iacobini, rotați, toboșari, jucători). Indicați penultima soluție generată.
- a. (toboșari, rotați, creți, iacobini, jucători)  
b. (toboșari, rotați, creți, jucători, iacobini)  
☒ c. (toboșari, rotați, jucători, creți, iacobini)  
d. (toboșari, rotați, jucători, iacobini, creți)
3. Fiecare dintre variabilele  $A$  și  $B$ , declarate alăturat, memorează coordonatele pozitive ( $x$  abscisa, iar  $y$  ordonata) ale câte unui punct în sistemul de coordonate  $xOy$ , extremități ale unui segment. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă cel puțin una dintre extremitățile segmentului precizat este în originea sistemului de coordonate  $xOy$ .
- |   |  |
|---|--|
| <pre>struct punct<br/>{ int x,y;<br/>} A,B;</pre> |  |
|---|--|
- a.  $(A.x + A.y) * (B.x + B.y) == 0$                       b.  $(A(x) + A(y)) * (B(x) + B(y)) == 0$   
c.  $(x.A + y.A) * (x.B + y.B) == 0$                       d.  $punct.A(x+y) * punct.B(x+y) == 0$
4. Într-un arbore cu rădăcină un nod se află pe nivelul  $x$  dacă lanțul elementar care are o extremitate în nodul respectiv și cealaltă extremitate în rădăcina arborelui are lungimea  $x$ . Pe nivelul 0 se află un singur nod (rădăcina).  
Un arbore cu rădăcină are 8 noduri, numerotate de la 1 la 8, și muchiile [1,3], [1,7], [1,8], [2,4], [3,5], [3,6], [4,5]. Știind că rădăcina arborelui este nodul numerotat cu 7, indicați numărul de niveluri ale arborelui dat.
- a. 3                      b. 4                      c. 6                      d. 7
5. Un graf orientat cu 5 vârfuri, numerotate de la 1 la 5, are arcele (1,4), (3,5), (5,1), (5,2). Indicați numărul minim de arce care trebuie adăugate acestuia, astfel încât graful obținut să fie tare conex.
- a. 1                      b. 2                      c. 3                      d. 4

**SUBIECTUL al II-lea**

**(40 de puncte)**

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

- Scrieți numărul afișat în urma executării algoritmului dacă pentru  $n$  se citește valoarea 5. **(6p.)**
- Scrieți două numere din intervalul  $[10, 10^2)$  care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 14. **(6p.)**
- Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat prima structură **pentru...execută** cu o structură repetitivă de alt tip. **(6p.)**

```

citește n (număr natural)
nr ← 0
pentru i ← n, 1, -1 execută
    x ← 0; y ← 1
    pentru j ← 1, i execută
        r ← 2 * x - y; x ← y; y ← r
    dacă y > 0 atunci
        nr ← nr + 1
scrie nr

```

- Subprogramul  $f$  este definit alăturat. Scrieți două numere naturale din intervalul  $[1, 10]$ , care pot fi memorate în variabilele întregi  $x_1$ , respectiv  $x_2$ , astfel încât valoarea lui  $f(10, x_1)$  să fie 5, iar valoarea lui  $f(x_2, 10)$  să fie 1. **(6p.)**

```

int f(int x, int y)
{
    if (x > y) return x * y + f(x - y, y);
    if (x < y) return y * x + f(x, y - x);
    return 1;
}

```

- Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 4 linii și 5 coloane, numerotate începând de la 0, cu elemente numere întregi, inițial toate nule. Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni astfel încât, în urma executării acesteia, variabila  $a$  să memoreze tabloul alăturat. **(6p.)**

|   |   |    |    |    |
|---|---|----|----|----|
| 1 | 5 | 9  | 13 | 17 |
| 2 | 6 | 10 | 14 | 18 |
| 3 | 7 | 11 | 15 | 19 |
| 4 | 8 | 12 | 16 | 20 |

**SUBIECTUL al III-lea**

**(30 de puncte)**

**1. Subprogramul `divPrimMax` are doi parametri:**

- $n$ , prin care primește un număr natural ( $n \in [2, 10^9]$ );
- $p$ , prin care furnizează cel mai mare divizor prim al lui  $n$ .

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă  $n=2000$ , în urma apelului  $p=5$ , deoarece  $2000=2^4 \cdot 5^3$ .

**(10p.)**

- Într-un text cu cel mult 100 de caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat și afișează pe ecran numărul de cuvinte ale sale formate dintr-un număr egal de vocale și consoane. Se consideră vocale literele din mulțimea  $a, e, i, o, u$ .

**Exemplu:** pentru textul

cuvantul consoane are un numar de patru vocale si patru consoane  
se afișează pe ecran 6.

**(10p.)**

- Se citesc de la tastatură două numere naturale din intervalul  $[1, 81]$ ,  $p_1$  și  $p_2$ , și se cere scrierea în fișierul `bac.out` a tuturor numerelor naturale cu exact 7 cifre, pentru care produsul primelor două cifre este egal cu  $p_1$ , cele trei cifre din mijloc sunt egale între ele, iar produsul ultimelor două cifre este egal cu  $p_2$ . Numerele apar în fișier în ordine strict crescătoare, fiecare pe câte o linie. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă  $p_1=12$ , iar  $p_2=8$ , atunci 2633324 și 3400018 sunt două dintre cele 160 de numere cu proprietatea cerută ( $2 \cdot 6=3 \cdot 4=12$  și  $2 \cdot 4=1 \cdot 8=8$ ).

- Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

- Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**