

Examenul de bacalaureat național 2020

Proba E. d)
Informatică
Limbajul C/C++

Testul 4

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. Variabile x , y și z sunt de tip întreg. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă x , y și z au valori identice.

a. $x==y \ \&\& \ x==z$

b. $x==y==z$

c. $!(x!=y \ \&\& \ x!=z)$

d. $!(x!=y!=z)$

2. Subprogramul f este definit alăturat. Pentru apelul $f(20, 2020)$, funcția se execută de:
- ```
int f(int x, int y)
{
 if(x<=1 || y<=1) return 0;
 if(x>y) return 1+f(f(x/y,y), y);
 return 1+f(x, f(x,y/x));
}
```

a. 5 ori

b. 9 ori

c. 11 ori

d. 20 de ori

3. Utilizând metoda backtracking, se generează toate modalitățile de a pregăti clătite, folosind, într-o anumită ordine, toate ingredientele din mulțimea {făină, lapte, ouă} pentru aluat, apoi unul dintre ingredientele din mulțimea {ciocolată, dulceață, urdă} pentru umplutură, și, la final, unul dintre ingredientele din mulțimea {cașcaval, mărar, frișcă} pentru ornare, având în vedere următoarele restricții: frișca se poate folosi numai împreună cu ciocolata și dulceața, iar mărarul și cașcavalul numai împreună cu urda. Primele cinci soluții generate sunt, în această ordine: (făină, lapte, ouă, ciocolată, frișcă), (făină, lapte, ouă, dulceață, frișcă), (făină, lapte, ouă, urdă, cașcaval), (făină, lapte, ouă, urdă, mărar), (făină, ouă, lapte, ciocolată, frișcă). Indicați a șaptea soluție generată.

a. (ouă, lapte, făină, urdă, mărar)

b. (lapte, făină, ouă, ciocolată, frișcă)

c. (făină, ouă, lapte, dulceață, frișcă)

d. (făină, ouă, lapte, urdă, cașcaval)

4. Numim înălțimea a unui arbore cu rădăcină numărul de muchii traversate de cel mai lung lanț elementar cu una dintre extremități în rădăcina arborelui. Un arbore cu rădăcină, cu 8 noduri, numerotate de la 1 la 8, este reprezentat prin vectorul "de tați" (6, 6, 5, 3, 0, 5, 8, 4). Indicați înălțimea arborelui:

a. 2

b. 3

c. 4

d. 5

5. Un graf neorientat are 10 noduri, numerotate de la 1 la 10, și muchiile [1,2], [1,3], [1,10], [3,10], [4,5], [4,6], [4,8], [5,7], [5,9], [6,8], [6,9], [7,9], [8,9]. Indicați numărul minim de muchii care trebuie adăugate pentru ca graful obținut să aibă cel puțin un lanț eulerian (lanț care traversează toate muchiile grafului).

a. 1

b. 2

c. 3

d. 4

**SUBIECTUL al II-lea**

**(40 de puncte)**

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

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

**a. Scrieți ce se afișează în urma executării algoritmului dacă se citește numărul 49335. (6p.)**

**b. Scrieți trei numere de patru cifre care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, valoarea afișată să fie 1100. (6p.)**

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

**d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structura `cât timp...execută` cu o structură repetitivă de alt tip. (6p.)**

```

citește n (număr natural)
repetă
 c1 ← n%10; n ← [n/10]; c2 ← n%10
 dacă c1 > c2 atunci
 c2 ← c1; c1 ← n%10
 ■
 cât timp c1 < c2 execută
 scrie c1
 c2 ← [c2/2]
 ■
până când n ≤ 9

```

**2. Variabila  $p$  memorează date despre un poliedru regulat: numărul de vârfuri, lungimea muchiei și două unghiuri specifice (dintre o față și o muchie, respectiv dintre două fețe). Știind că expresiile C/C++ de mai jos au ca valori un număr natural reprezentând numărul de vârfuri ale poliedrului, și numerele reale reprezentând lungimea muchiei, respectiv cele două unghiuri specifice, scrieți definiția unei structuri cu eticheta `poliedru`, care permite memorarea datelor precizate, și declarați corespunzător variabila  $p$ .**

`p.NrVarfuri`      `p.Muchie`      `p.Unghi.FataMuchie`      `p.Unghi.FataFata`      **(6p.)**

**3. Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $s$  poate memora un șir de cel mult 20 de caractere. Scrieți șirul memorat de variabila  $s$  în urma executării secvenței de mai jos.**

```

strcpy(s, "optsprezece"); i=0; j=strlen(s)-1;
while (i < j)
{
 if (strchr("aeiou", s[i]) == NULL && strchr("aeiou", s[j]) != NULL)
 {
 s[i] = s[i]+1; s[j] = s[j]-1;
 i=i+1; j=j-1;
 }
}

```

**(6p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

**1. Două numere  $a$  și  $b$  sunt numite **generatoare** ale unui număr natural  $n$  dacă  $a \cdot b + [a/b] = n$ , unde s-a notat cu  $[c]$  partea întreagă a numărului real  $c$ .**

Subprogramul **generatoare** are un singur parametru,  $n$ , prin care primește un număr natural ( $n \in [2, 10^9]$ ). Subprogramul afișează pe ecran toate perechile distincte de numere naturale cu proprietatea că sunt generatoare ale lui  $n$  și că primul număr din pereche este par. Numerele din fiecare pereche sunt separate prin simbolul minus (-), iar perechile sunt separate prin câte un spațiu. Dacă nu există astfel de perechi, se afișează pe ecran mesajul **nu exista**. Scrieți definiția completă a subprogramului.

**Exemplu:** dacă  $n=2020$  se afișează pe ecran

2-1010 4-505 10-202 20-101 96-21 200-10 606-3 808-2 1010-1

**(10p.)**

**2. Într-un tablou bidimensional, cu elemente având valori numai în mulțimea  $\{0, 1\}$ , numim coloane „complementare” două coloane cu proprietatea că oricare două elemente ale acestora, aflate pe aceeași linie, sunt diferite.**

Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 20]$ ,  $m$  și  $n$ , și elementele unui tablou bidimensional cu  $m$  linii și  $n$  coloane, numere naturale din mulțimea  $\{0, 1\}$ . Programul afișează pe ecran numărul de coloane ale tabloului care sunt „complementare” cu prima coloană a acestuia.

**Exemplu:** dacă  $m=3$ ,  $n=6$ , pentru tabloul alăturat se afișează pe ecran 3.

**(10p.)**

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |

**3. Fișierul `bac.txt` conține, în ordine descrescătoare, cel puțin două și cel mult  $10^6$  numere naturale din intervalul  $[0, 10^9]$ , separate prin câte un spațiu. Se cere să se afișeze pe ecran, în ordine strict descrescătoare, separate prin câte un spațiu, numai numerele care apar în fișier de exact două ori. Dacă nu există niciun astfel de număr, se afișează pe ecran mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.**

**Exemplu:** dacă fișierul conține numerele 100 50 50 50 49 49 36 16 16 12 10 10 9 7 7

pe ecran se afișează, în această ordine, numerele 49 16 10 7

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

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