

Examenul de bacalaureat național 2020

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

Testul 12

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. Variabilele **x** și **y** sunt întregi și memorează câte un număr natural. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în **x** este strict mai mare decât 0 și numărul memorat în **y** este strict mai mare decât 20.

- |                                       |  |
|---------------------------------------|--|
| a. <b><code>x*y-20!=0</code></b>      | b. <b><code>x*(y-20)!=0</code></b>       |
| c. <b><code>x*(y-20)&gt;=0</code></b> | d. <b><code>!(x*(y-20)&lt;=0)</code></b> |

2. Indicați o declarație a unui tablou bidimensional cu **exact 40** de elemente, numere întregi cu cel mult 2 cifre.

- |                                      |                                     |
|--------------------------------------|-------------------------------------|
| a. <b><code>int m[20][2];</code></b> | b. <b><code>float m[80];</code></b> |
| c. <b><code>int m[2][40];</code></b> | d. <b><code>int m[80];</code></b>   |

3. Subprogramul **f** este definit alăturat. Indicați ce se va afișa în urma apelului de mai jos.  
**f(54321);**

```
void f(int n)
{ if (n!=0)
  { f(n/10);
    if (n%2==0)
      cout<<n%10; | printf("%d",n%10);
  }
}
```

- |              |              |               |               |
|--------------|--------------|---------------|---------------|
| a. <b>24</b> | b. <b>42</b> | c. <b>315</b> | d. <b>513</b> |
|--------------|--------------|---------------|---------------|

4. Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6, este reprezentat prin matricea de adiacență alăturată. Indicați arcele care trebuie eliminate, astfel încât în graful parțial obținut fiecare vârf să aibă gradul exterior egal cu gradul interior.

0	1	0	1	0	0
0	0	1	0	0	1
0	0	0	1	0	0
1	1	0	0	0	0
0	0	0	1	0	0
0	0	1	0	1	0

- |                        |                        |                        |                        |
|------------------------|------------------------|------------------------|------------------------|
| a. <b>(1,4), (3,4)</b> | b. <b>(1,4), (6,3)</b> | c. <b>(3,4), (5,4)</b> | d. <b>(4,2), (6,3)</b> |
|------------------------|------------------------|------------------------|------------------------|

5. Un graf neorientat cu 6 noduri, numerotate de la 1 la 6, are muchiile [1,2], [1,3], [2,3], [2,4], [2,6], [3,4], [3,5], [4,5], [4,6]. Indicați numărul **minim** de muchii ce pot fi eliminate, astfel încât graful parțial obținut să nu mai fie conex.

- |             |             |             |             |
|-------------|-------------|-------------|-------------|
| a. <b>4</b> | b. <b>3</b> | c. <b>2</b> | d. <b>1</b> |
|-------------|-------------|-------------|-------------|

**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 numărul afișat în urma executării algoritmului dacă se citește valoarea 2592. **(6p.)**

b. Scrieți trei numere din intervalul  $[10^3, 10^4)$  care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 20. **(6p.)**

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

citește  $n$  (număr natural nenul)

$nr \leftarrow 0$

$c \leftarrow 9$

cât timp  $c \geq 0$  execută

$m \leftarrow n$

cât timp  $m \neq 0$  și  $m \% 10 \neq c$  execută

$m \leftarrow [m/10]$

■

dacă  $m \neq 0$  atunci

$nr \leftarrow nr * 10 + m \% 10$

■

$c \leftarrow c - 1$

■

scrie  $nr$

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

2. Variabila  $f$  memorează simultan următoarele date despre fiecare dintre cele 20 de specii de flori dintr-o florărie: un cod, reprezentând specia, numărul de varietăți ale speciei respective și numărul de exemplare din fiecare dintre acestea. În florărie sunt maximum 10 varietăți din fiecare specie. Știind că expresiile C/C++ de mai jos au ca valori numere naturale și reprezintă codul și numărul de varietăți ale primei specii, respectiv numărul de exemplare din cea de a 4-a varietate din această specie, scrieți definiția unei structuri cu eticheta `flori`, care permite memorarea datelor despre o specie de flori, și declarați corespunzător variabila  $f$ .

`f[0].cod`      `f[0].nrVarietati`      `f[0].nrExemplare[3]` **(6p.)**

3. Utilizând metoda backtracking, se generează toate numerele de câte 5 cifre, cu proprietatea că aceste cifre sunt în ordine strict crescătoare, iar cifra unităților este 6. Scrieți ultimele trei numere generate. **(6p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Subprogramul `pDoi` are un singur parametru,  $n$ , prin care primește un număr natural ( $n \in [1, 10^9]$ ). Subprogramul returnează cea mai mare valoare din intervalul  $[1, n]$ , cu proprietatea că este o putere a lui 2.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă  $n=20$ , subprogramul returnează 16. **(10p.)**

2. Într-un text cu cel mult  $10^2$  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 precizat, pe care îl transformă în memorie, eliminând numai ultima vocală care apare în text, ca în exemplu. Programul afișează pe ecran textul obținut sau mesajul `nu exista`, dacă în text nu există nicio vocală. Se consideră vocale literele `a, e, i, o, u`.

**Exemplu:** dacă se citește textul: `cuvantul ritm poate fi tradus rhythm`

se obține textul `cuvantul ritm poate fi trads rhythm` **(10p.)**

3. Șirul de mai jos este definit alăturat:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 21, 42, 43, 86....

$$f_n = \begin{cases} n & \text{dacă } n \leq 10 \\ 2 \cdot f_{n-1} & \text{dacă } n > 10 \text{ și } n \text{ impar} \\ 1 + f_{n-1} & \text{altfel} \end{cases}$$

Se citește de la tastatură un număr natural  $x$  ( $x \in [1, 10^9]$ ), termen al șirului dat, și se cere să se scrie în fișierul `bac.out`, separați prin câte un spațiu, în ordine strict descrescătoare, toți termenii șirului mai mici sau egali cu  $x$ . Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

**Exemplu:** dacă valoarea citită de la tastatură este 43

se scriu în fișier numerele 43 42 21 20 10 9 8 7 6 5 4 3 2 1

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

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