

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

**Testul 17**

*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. Variabila **x** este de tip întreg și memorează un număr nenul. Indicați expresia C/C++ cu valoarea 1 pentru orice multiplu al lui 2020 memorat în variabila **x**.

a.  **$x / (x / 2020) == 0$**       b.  **$x / (x \% 2020) == 0$**       c.  **$x \% (x \% 2020) == 0$**       d.  **$x \% (x / 2020) == 0$**

2. Subprogramul **f** este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.  
**f**(2,20);

```
void f(int x, int y)
{
    if(x < y)
        f(2*x-1, y-1);
    cout << x+y << " "; | printf("%d ", x+y);
}
```

a. **22 22 23 26**      b. **22 22 23 26 33**      c. **26 23 22 22**      d. **33 26 23 22 22**

3. Utilizând metoda backtracking se generează toate posibilitățile de a forma liste de câte 3 locuri izolate distincte din lume, din mulțimea {**Hanging**, **Meteora**, **Sumela**, **Taktsang**, **Taung Kalat**}, astfel încât pe oricare două poziții alăturate să nu se afle locuri din submulțimea {**Hanging**, **Sumela**, **Taung Kalat**}. Două liste diferă prin cel puțin un loc sau prin ordinea acestora. Primele șase soluții generate sunt, în această ordine: (**Hanging**, **Meteora**, **Sumela**), (**Hanging**, **Meteora**, **Taktsang**), (**Hanging**, **Meteora**, **Taung Kalat**), (**Hanging**, **Taktsang**, **Meteora**), (**Hanging**, **Taktsang**, **Sumela**), (**Hanging**, **Taktsang**, **Taung Kalat**). Indicați numărul de soluții generate care au pe prima poziție **Meteora**.

a. **4**      b. **6**      c. **8**      d. **10**

4. Un arbore cu rădăcină are 8 noduri, numerotate de la 1 la 8, și este reprezentat prin vectorul de „tați” (5, 7, 5, 6, 8, 5, 8, 0). Indicați frunzele arborelui.

a. **1, 2, 3, 4**      b. **1, 2, 3**      c. **1, 2, 6**      d. **4**

5. Un graf orientat cu 5 vârfuri este reprezentat prin matricea de adiacență alăturată. Indicați numărul de vârfuri ale unui subgraf al acestuia care are un număr maxim de vârfuri izolate.

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

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  $a \% b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$ .

**a. Scrieți ce se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele 3 și 12. (6p.)**

**b. Scrieți două seturi de date din intervalul  $[1, 10]$  care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze numărul 20. (6p.)**

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

```
citește x,y
    (numere naturale  $x \leq y$ )
i ← x; j ← y; s ← 0
cât timp i ≤ j execută
    dacă i % 2 = 0 atunci
        s ← s + j
    ■
    dacă j % 2 = 0 atunci
        s ← s + i
    ■
    i ← i + 1; j ← j - 1
    ■
scrie s
```

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

**2. Variabila *ob* memorează simultan următoarele date despre un obiectiv turistic din evidența unei agenții: un cod și o perioadă din an în care se recomandă să fie vizitat, formată din zile consecutive, între două date (ziua și luna de început, respectiv ziua și luna finală). Expresiile C/C++ de mai jos au ca valori numere naturale și reprezintă codul obiectivului, ziua și luna de început, respectiv ziua și luna finală a perioadei recomandate pentru vizitarea acestui obiectiv. Scrieți definiția unei structuri cu eticheta *obiectiv*, care permite memorarea datelor despre un obiectiv turistic, și declarați corespunzător variabila *ob*.**

*ob.cod ob.dataInceput.zi ob.dataInceput.luna ob.dataFinal.zi ob.dataFinal.luna*  
**(6p.)**

**3. 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 C/C++ astfel încât, în urma executării acesteia, variabila *a* să memoreze tabloul alăturat. (6p.)**

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

**SUBIECTUL al III-lea**

**(30 de puncte)**

**1. Subprogramul *maxim* are un singur parametru, *n*, prin care primește un număr natural ( $n \in [0, 10^9]$ ). Subprogramul returnează cea mai mare cifră impară din scrierea acestuia, sau -1 dacă nu există astfel de cifre. Scrieți definiția completă a subprogramului.**

**Exemplu:** dacă  $n = \underline{5672883}$ , subprogramul returnează 7.

**(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 menționat, pe care îl modifică în memorie, duplicând fiecare cuvânt format numai din vocale. Cuvântul duplicat este separat prin câte un spațiu de cuvintele vecine. Textul transformat este afișat pe ecran, iar dacă nu există astfel de cuvinte, se afișează pe ecran mesajul *nu exista*.**

**Exemplu:** dacă textul citit este *oaia aia alba e a ei*

se obține textul *oaia oaia aia aia alba e e a a ei ei*

**(10p.)**

**3. Fișierul text *bac.txt* conține numere naturale din intervalul  $[1, 10^4]$ : pe prima linie un număr *n*, pe a doua linie un șir de *n* numere, iar pe fiecare dintre următoarele linii, până la finalul fișierului, câte o pereche de numere, reprezentând extremitățile unui interval închis. Numerele aflate pe aceeași linie a fișierului sunt în ordine crescătoare și sunt separate prin câte un spațiu.**

Se cere să se afișeze pe ecran numărul de intervale care nu conțin niciun termen al șirului aflat pe a doua linie a fișierului. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

**Exemplu:** dacă fișierul conține numerele alăturate, se afișează pe ecran 3.

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

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

5
4 8 9 16 25
<u>1 3</u>
2 5
9 15
<u>5 7</u>
20 100
<u>10 12</u>