

Simulare județeană
Examenul național de bacalaureat național 2025
Proba E.d)
INFORMATICĂ

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. Expresia C/C++

$!(x < 2001) \ \&\& \ (x \leq 2002) \ || \ (x \geq 2019) \ \&\& \ !(x > 2020)$

are valoarea 1 dacă și numai dacă valoarea memorată de variabila reală x aparține reuniunii:

- a. $\{2001\} \cup [2002, 2019] \cup \{2020\}$ b. $[2001, 2002] \cup \{2019\} \cup \{2020\}$
c. $[2001, 2002] \cup \{2019, 2020\}$ d. $[2001, 2002] \cup [2019, 2020]$

2. Subprogramul f este definit alăturat. `int f (int x)`

Indicați valoarea $f(102030)$.

```
{  
    if(x>20)  
        return 20+f(x/10);  
    return 2020;  
}
```

- a. 1010 b. 2020 c. 2100 d. 3200

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. Un arbore cu 10 noduri, numerotate de la 1 la 10, este reprezentat prin vectorul de „tați” $t=(2,5,1,1,0,3,3,7,4,6)$. Indicați numărul de frunze ale arborelui.

- a. 4 b. 3 c. 5 d. 6

5. Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6, are arcele (1,6), (2,1), (2,3), (2,5), (3,4), (3,5), (4,6). Indicați numărul minim de arce care pot fi adăugate astfel încât graful obținut să aibă cel puțin două circuite. Un circuit este format numai din arce distincte, iar două circuite sunt distincte dacă diferă prin cel puțin un arc.
- 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 și cu $[c]$ partea întreagă a numărului real c .
- a. Scrieți valoarea afișată dacă se citesc, în această ordine, numerele 12, 7, 354, 9, 1630, 0. **(6p.)**
- b. Scrieți o secvență de numere pare care pot fi citite, în această ordine, astfel încât, în urma executării algoritmului, să se afișeze valoarea 751. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat prima structură cât timp... execută cu o structură repetitivă cu test final. **(6p.)**

```

citește x (număr natural)
n ← 0
cât timp x ≠ 0 execută
    y ← x; c ← 0
    cât timp y > 0 execută
        dacă y % 10 > c atunci
            c ← y % 10
        y ← [y / 10]
    n ← n * 10 + c
citește x (număr natural)
scrie n
    
```

2. Variabila r memorează simultan următoarele date despre fiecare dintre cele 20 de rase de porumbei dintr-o volieră: un cod, reprezentând rasa, numărul de exemplare din rasa respectivă și vârstele acestora. În volieră sunt maximum 10 exemplare din fiecare rasă. Știind că expresiile C/C++ de mai jos au ca valori numere naturale și reprezintă codul și numărul de exemplare din prima rasă, respectiv vârsta celui de al 4-lea exemplar din această rasă, scrieți definiția unei structuri cu eticheta **rasa**, care permite memorarea datelor despre o rasă de porumbei, și declarați corespunzător variabila r .
- $r[0].cod$ $r[0].nrExemplare$ $r[0].varsta[3]$ **(6p.)**
3. Variabila s poate memora un șir de cel mult 20 de caractere. Scrieți ce se afișează în urma executării secvenței alăturate. **(6p.)**
- ```

strcpy(s, "militar");
cout << s + 4 << endl; | printf("%s\n", s + 4);
s[0] = s[0] - 1; s[1] = s[0] - 3;
s[2] = s[0] + 1; s[3] = '\0';
cout << s; | printf("%s", s);

```

**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  $n=56792883$ , subprogramul returnează 9  
(10p.)

2. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 20]$ ,  $n$  și  $k$ , și construiește în memorie un tablou bidimensional cu  $n$  linii și  $n \cdot k$  coloane, numerotate începând cu 1, astfel încât fiecare linie  $i$  ( $i \in [1, n]$ ) memorează un șir crescător de termeni cu proprietatea că primul termen este  $i$ , fiecare valoare apare în șir de exact  $k$  ori și oricare doi termeni alăturați au valori egale sau consecutive.

Programul afișează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu valorile aflate pe aceeași linie separate prin câte un spațiu.

**Exemplu:** dacă  $n=4$  și  $k=3$ , se afișează pe ecran tabloul alăturat.  
(10p.)

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 |
| 2 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 |
| 3 | 3 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 6 |
| 4 | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 | 7 |

3. 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  $2 \cdot 4 = 1 \cdot 8 = 8$ ).

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

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