

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

Testul 7

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.

- Expresia C/C++  
`!(x<2001) && (x<=2002) || (x>=2019) && !(x>2020)`  
are valoarea 1 dacă și numai dacă valoarea memorată de variabila reală  $x$  aparține reuniunii:
  - $\{2001\} \cup [2002, 2019] \cup \{2020\}$
  - $[2001, 2002] \cup \{2019\} \cup \{2020\}$
  - $[2001, 2002] \cup \{2019, 2020\}$
  - $[2001, 2002] \cup [2019, 2020]$
- Subprogramul `f` este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.  
`f(12345);`

```
void f (int x)
{ cout<<"+"; | printf("+");
  if (x>0)
  { f(x/100);
    cout<<x; | printf("%d", x);
  }
  cout<<"+"; | printf("+");
}
```

  - `+++++1+123+12345+`
  - `++++112312345`
  - `+12345+123+1+++++`
  - `+1+123+12345+`
- Utilizând metoda backtracking se generează toate modalitățile de forma un grup de patru arbori cu flori din mulțimea {**albizia**, **jacaranda**, **laburnum**, **magnolie**, **mimoza**, **sakura**}. După grupuri diferite prin cel puțin un arbore. Primele cinci soluții generate sunt în această ordine: (**albizia**, **jacaranda**, **laburnum**, **magnolie**), (**albizia**, **jacaranda**, **laburnum**, **mimoza**), (**albizia**, **jacaranda**, **laburnum**, **sakura**), (**albizia**, **jacaranda**, **magnolie**, **mimoza**), (**albizia**, **jacaranda**, **magnolie**, **sakura**). Indicați o enumerare care este generată ca soluție.
  - (**magnolie**, **mimoza**, **laburnum**, **sakura**)
  - (**jacaranda**, **magnolie**, **mimoza**, **sakura**)
  - (**jacaranda**, **laburnum**, **magnolie**, **mimoza**, **sakura**)
  - (**albizia**, **jacaranda**, **magnolie**, **laburnum**)
- Un graf orientat cu 5 vârfuri este reprezentat prin matricea de adiacență alăturată. Indicați numărul vârfurilor cu gradul interior 2.

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

  - 1
  - 2
  - 3
  - 4
- Un graf neorientat are 20 de noduri și 5 componente conexe, fiecare dintre acestea fiind arbore. Indicați numărul de muchii ale grafului.
  - 7
  - 11
  - 15
  - 19

**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  $c$ .
 

<ol style="list-style-type: none"> <li>a. Scrieți ce se afișează dacă se citește numărul 253387. <b>(6p.)</b></li> <li>b. Scrieți cel mai mic și cel mai mare număr din intervalul <math>[10, 10^2)</math> care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 0. <b>(6p.)</b></li> <li>c. Scrieți programul C/C++ corespunzător algoritmului dat. <b>(10p.)</b></li> <li>d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat prima structură repetitivă cu o structură repetitivă de alt tip. <b>(6p.)</b></li> </ol>	<pre> citește n (număr natural nenul) x ← 1; m ← 0; p ← 1 cât timp x &lt; 10 execută     cn ← n     cât timp cn ≠ 0 execută         c ← cn % 10; cn ← [cn / 10]         dacă c = x atunci             m ← c * p + m; p ← p * 10     x ← x + 2 scrie m           </pre>
--	--
2. Variabila  $c$  memorează simultan numărul de cărți dintr-o bibliotecă (număr natural din intervalul  $[3, 10^2)$ ) și date despre fiecare carte (titlu și autor, șiruri de cel mult 20 de caractere). Expresiile C/C++ de mai jos au ca valori numărul de cărți, titlul și numele autorului celei de a treia cărți. Scrieți definiția unei structuri cu eticheta `biblio`, care permite memorarea datelor despre o bibliotecă, și declarați corespunzător variabila  $c$ .
 

<code>c.numar</code> <code>c.carte[2].titlu</code> <code>c.carte[2].autor</code>	<b>(6p.)</b>
--	--------------
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, "stilou");
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] = s[0] + 3;
s[4] = '\0';
cout << s; | printf("%s", s);
          
```

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Subprogramul `putere` are doi parametri,  $n$  și  $p$ , prin care primește câte un număr natural ( $n \in [2, 10^9]$ ,  $p \in [0, 10^9]$ ). Subprogramul returnează puterea la care apare numărul  $p$  în descompunerea în factori primi a lui  $n$ , dacă  $p$  este număr prim, sau valoarea  $-1$  în caz contrar. Scrieți definiția completă a subprogramului.  
**Exemplu:** dacă  $n=80$  și  $p=2$ , subprogramul returnează numărul 4 ( $80=2^4 \cdot 5$ ). **(10p.)**
2. O valoare  $k$  **polarizează** două șiruri dacă există doi termeni care au acea valoare, unul fiind în primul șir, iar celălalt în al doilea șir.  
Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul  $[1, 20]$ :  $m$ ,  $n$  și elementele unui tablou bidimensional cu  $m$  linii și  $n$  coloane, cu proprietatea că nu există două elemente egale situate pe aceeași linie sau pe aceeași coloană. Programul afișează pe ecran valorile care pot polariza două șiruri, și anume șirul format din elementele de pe prima coloană, respectiv șirul format din elementele ultimei coloane a tabloului.  
Valorile sunt afișate într-o ordine oarecare, separate prin câte un spațiu, iar dacă nu există astfel de valori se afișează pe ecran mesajul **nepolarizate**.  
**Exemplu:** pentru  $m=4$ ,  $n=5$  și tabloul alăturat se afișează pe ecran, nu neapărat în această ordine, numerele  $5 \ 6$  **(10p.)**

3	7	1	2	5
2	4	5	9	6
6	2	7	8	1
5	3	2	7	8
3. Fișierul `bac.txt` conține un șir de cel mult  $10^6$  numere întregi din intervalul  $[-10^3, 10^3]$ , separate prin câte un spațiu. Se cere să se afișeze pe ecran suma minimă obținută adunând numere de pe poziții consecutive în șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.  
**Exemplu:** dacă fișierul `bac.txt` conține valorile -4 6 -7 -2 1 -4 10 3 -9 -2 2 se afișează pe ecran numărul -12
  - a. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
  - b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**