

Examenul național de bacalaureat 2021

**Proba E. d)
INFORMATICĂ
Limbajul C/C++**

Testul 10

*Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică*

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei 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 muchie are extremități distincte și oricare două 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++
`!(a<2001) && (a<=2002) || (a>=2020) && !(a>2021)`
are valoarea 1 dacă și numai dacă valoarea memorată de variabila reală a aparține:
 - $[2001, 2002] \cap \{2020, 2021\}$
 - $[2001, 2002] \cup [2020, 2021]$
 - $\{2001\} \cup [2002, 2020] \cup \{2021\}$
 - $[2001, 2002] \cap \{2020\} \cap \{2021\}$
- Subprogramul `afis` este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.
`afis(12345);`

```
void afis (int a)
{ cout<<"+"; | printf("+");
  if(a>0)
  { afis(a/1000);
    cout<<a; | printf("%d", a);
  }
  cout<<"+"; | printf("+");
}
```

 - `+12345+123+1+`
 - `+1+123+12345+`
 - `++++12+12345+`
 - `++++112312345`
- Utilizând metoda backtracking, se generează toate modalitățile de forma un grup de patru persoane din mulțimea **{Ana, Ioana, Lia, Maria, Miruna, Simona}**. Două grupuri diferă prin cel puțin o persoană. Primele cinci soluții generate sunt, în această ordine: **(Ana, Ioana, Lia, Maria)**, **(Ana, Ioana, Lia, Miruna)**, **(Ana, Ioana, Lia, Simona)**, **(Ana, Ioana, Maria, Miruna)**, **(Ana, Ioana, Maria, Simona)**. Indicați o enumerare generată ca soluție, în această formă.
 - (Ana, Lia, Maria, Miruna, Simona)**
 - (Ioana, Maria, Miruna, Simona)**
 - (Lia, Ioana, Maria, Simona)**
 - (Maria, Miruna, Lia, Simona)**
- Un graf neorientat are 6 noduri, numerotate de la 1 la 6, și muchiile **[1,2], [1,3], [2,3], [2,4], [2,5], [2,6], [3,4], [4,5]**. Indicați numărul nodurilor care au gradul un număr par.
 - 5
 - 4
 - 3
 - 2
- Un graf neorientat are 20 de noduri și 4 componente conexe, fiecare dintre acestea fiind arbore. Indicați numărul de muchii ale grafului.
 - 7
 - 11
 - 16
 - 19

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural, nenul, y .

a. Scrieți ce se afișează dacă se citesc, în această ordine, valorile 2 și 9. **(6p.)**

b. Dacă pentru x se citește valoarea 3, scrieți două numere din intervalul $[1, 20]$ care pot fi citite pentru y astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, rezultatul afișat să fie format din exact 12 caractere. **(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 x,y (numere naturale)
┌dacă x<y atunci
| x←x-y; y←x+y; x←y-x
└─
┌cât timp x≥y execută
| scrie 'A'
| x←x-y
| ┌dacă x%2=0 atunci scrie 'A'
| └altfel scrie 'B'
└─
└─

```

2. Variabila y memorează simultan datele fiecăruia dintre cei 30 de elevi dintr-o clasă: codul de identificare la un examen (un număr natural din intervalul $[1, 10^3]$) și două note obținute (numere reale). Expresiile C/C++ de mai jos au ca valori codul de identificare și cele două note ale celui de al treilea elev din clasă. Scrieți definiția unei structuri cu eticheta `elev`, care permite memorarea datelor despre un elev, și declarați corespunzător variabila y .

`y[2].cod` `y[2].nota1` `y[2].nota2` **(6p.)**

3. Scrieți ce se afișează în urma executării secvenței de mai jos, în care variabilele s și t permit memorarea câte unui șir de cel mult 50 de caractere.

```

strcpy(s, "vorbeste");
s[3]=s[0]; s[5]=s[2]; s[0]=s[1]+1; s[2]=s[1]-2; s[6]=s[4]-1;
strcpy(t, s); t[3]='\0';
cout<<t<<endl<<s+3; | printf("%s\n%s", t, s+3);

```

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Numerele naturale x și y sunt numite în armonie dacă suma lor aparține intervalului deschis definit de suma divizorilor lui x , respectiv suma divizorilor lui y .

Subprogramul `armonie` are doi parametri, x și y , prin care primește câte un număr natural din intervalul $[1, 10^6]$. Subprogramul returnează valoarea 1, dacă x și y sunt în armonie, sau valoarea 0 în caz contrar.

Scrieți definiția completă a subprogramului.

Exemplu: dacă $x=8$, iar $y=12$ subprogramul returnează 1 ($1+2+4+8=15$, $1+2+4+6+12=25$, iar $8+12=20 \in (15, 25)$), iar dacă $x=8$ și $y=13$, subprogramul returnează 0 ($1+2+4+8=15$, $1+13=14$, iar $8+13=21 \notin (14, 15)$). **(10p.)**

2. O valoare **filtrează** 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 $[2, 20]$: 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 filtra șirul format din primele $n-1$ elemente de pe prima linie, respectiv șirul format din ultimele $m-1$ elemente ale ultimei coloane a tabloului, ca în exemplu. Valorile sunt afișate într-o ordine oarecare, separate prin câte un spațiu, sau mesajul `nu exista`, dacă nu există astfel de valori.

Exemplu: pentru $m=5$, $n=4$ și tabloul alăturat, se afișează pe ecran, nu neapărat în această ordine, numerele 4 7 **(10p.)**

4	5	7	2
2	7	3	6
7	6	4	0
6	9	8	7
8	0	5	4

3. Fișierul `bac.txt` conține un șir de cel mult 10^5 numere naturale din intervalul $[1, 10^9]$, separate prin câte un spațiu.

Se cere să se afișeze pe ecran cea mai mică poziție pe care ar putea-o ocupa primul termen al șirului aflat în fișier în șirul format cu aceleași valori, ordonat crescător. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă fișierul conține numerele 15 7 15 17 6 4

se afișează pe ecran 4 (15 se află pe a patra și pe a cincea poziție în șirul 4, 6, 7, 15, 15, 17).

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

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