

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

Testul 18

*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.

- Variabila  $x$  este de tip întreg și memorează un număr nenul. Indicați expresia C/C++ cu valoarea 1 dacă  $x$  memorează un divizor al lui 2020.
  - $2020 / (2020 / x) == 0$
  - $2020 / (2020 \% x) == 0$
  - $2020 \% (2020 \% x) == 0$
  - $2020 \% (2020 / x) == 0$
- Variabila  $a$  memorează un tablou bidimensional cu 6 linii și 6 coloane, numerotate de la 0 la 5, cu elemente numere întregi, iar toate celelalte variabile sunt întregi.

```
for (i=0; i<6; i++)
for (j=0; j<6; j++)
a[i][j]=(i%2)*j+(j%2)*i;
```

Indicați valoarea sumei elementelor de pe diagonala principală a tabloului construit în urma executării secvenței de mai sus.
  - 6
  - 12
  - 18
  - 30
- Utilizând metoda backtracking se generează toate posibilitățile de a realiza o listă de 3 lucrări distincte ale lui George Enescu din mulțimea {Oedip, Poema română, Rapsodia română nr. 1, Rapsodia română nr. 2, Simfonia nr. 1}. Două liste sunt distincte dacă diferă prin cel puțin o lucrare sau prin ordinea acestora. Primele patru soluții generate sunt, în această ordine: (Oedip, Poema română, Rapsodia română nr. 1), (Oedip, Poema română, Rapsodia română nr. 2), (Oedip, Poema română, Simfonia nr. 1), (Oedip, Rapsodia română nr. 1, Poema română). Indicați penultima soluție generată.
  - (Simfonia nr. 1, Rapsodia română nr. 2, Poema română)
  - (Simfonia nr. 1, Rapsodia română nr. 2, Oedip)
  - (Rapsodia română nr. 2, Rapsodia română nr. 1, Poema română)
  - (Rapsodia română nr. 2, Simfonia nr. 1, Oedip)
- Un arbore cu 10 noduri, numerotate de la 1 la 10, este reprezentat prin vectorul de „tați” (5, 1, 10, 7, 0, 4, 1, 7, 4, 5). Indicați numărul total de descendenți ai nodului 7.
  - 1
  - 2
  - 3
  - 4
- Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6, este reprezentat prin matricea de adiacență alăturată. Precizați numărul tuturor grafurilor parțiale distincte ale grafului dat. Două grafuri parțiale sunt distincte dacă matricele lor de adiacență sunt diferite.

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

  - $2^6$
  - $2^8$
  - $2^{12}$
  - $2^{28}$

**SUBIECTUL al II-lea**

**(40 de puncte)**

1. Algoritmul alăturat este reprezentat în pseudocod.
- a. Scrieți ce se afișează dacă se citesc, în această ordine, numerele 2 și 3. (6p.)
- b. Scrieți două seturi de valori din intervalul  $[1, 10^2]$  care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, ultima valoare afișată să fie 20. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat una dintre structurile **pentru...execută** cu o structură repetitivă de alt tip. (6p.)
- ```

citește n, k
(numere naturale nenule)
pentru i ← 1, n execută
    pentru j ← k, 1, -1 execută
        scrie j, ' '
    ■
    pentru j ← 1, k execută
        scrie j, ' '
    ■
    k ← k - 1
    ■
    
```
2. Subprogramul  $f$  este definit alăturat. Scrieți ce valori au  $f(0)$ , respectiv  $f(23575209)$ . (6p.)
- ```

int f (int n)
{ if (n!=0) return n%2+f(n/100);
  return 0;
}
    
```
3. Un graf neorientat cu 9 noduri, numerotate de la 1 la 9, are muchiile  $[1, 2]$ ,  $[1, 6]$ ,  $[2, 6]$ ,  $[3, 7]$ ,  $[4, 5]$ ,  $[4, 9]$ . Scrieți numărul componentelor conexe ale grafului și câte dintre acestea au un număr maxim de noduri. (6p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Subprogramul **suma** are doi parametri:
- $n$ , prin care primește un număr natural din intervalul  $[0, 10^9]$ ;
  - $s$ , prin care furnizează suma cifrelor pare distincte din scrierea acestuia.
- Scrieți definiția completă a subprogramului.  
**Exemplu:** dacă  $n=67638825$ , după apel  $s=16$  ( $16=6+8+2$ ), iar dacă  $n=15$ , după apel  $s=0$ . (10p.)
2. Un cuvânt este **sufix** al unui alt cuvânt dacă se obține din acesta, prin eliminarea primelor sale litere. Scrieți un program C/C++ care citește de la tastatură două numere naturale  $n$  și  $k$  ( $n \in [2, 20]$ ,  $k \in [1, n]$ ) și apoi  $n$  cuvinte distincte, fiecare fiind format din cel mult 20 de caractere, numai litere mici ale alfabetului englez. La introducerea datelor, după fiecare cuvânt se tastează Enter. Programul afișează pe ecran, separate prin câte un spațiu, cuvintele care îl au drept sufix pe al  $k$ -lea cuvânt citit, ca în exemplu. Dacă nu există astfel de cuvinte, se afișează pe ecran mesajul **nu exista**.  
**Exemplu:** dacă  $n=7$ ,  $k=3$  și se citesc cuvintele alăturate, pe ecran se afișează **paratirisi hiritisi** (10p.)
- ```

isihast
paratirisi
isi
meremetisire
acolisitor
hiritisi
paraponisit
    
```
3. Numim **10-secvență** într-un șir de numere naturale, o succesiune de termeni aflați pe poziții consecutive în șir, cu proprietatea că sunt multipli ai numărului 10. **Lungimea** secvenței este egală cu numărul de termeni ai săi. Fișierul **bac.txt** conține un șir de cel mult  $10^6$  numere naturale din intervalul  $[0, 10^9]$ , separate prin câte un spațiu. Cel puțin un termen din șir este multiplu al lui 10. Se cere să se afișeze pe ecran două valori, separate printr-un spațiu, reprezentând lungimea maximă a unei 10-secvențe din șirul aflat în fișier, respectiv numărul de 10-secvențe cu o astfel de lungime. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.  
**Exemplu:** dacă fișierul are conținutul
- ```

7 3 200 100 10 9 6 100 1000 40 1002 20 30
    
```
- alăturat, se afișează 3 2
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)