

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

Testul 9

*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 și numai dacă 2020 este divizor al lui  $x$ .  
a.  $x / (x / 2020) == 0$       b.  $x / (x \% 2020) == 0$       c.  $x \% (x \% 2020) == 0$       d.  $x \% (x / 2020) == 0$
- 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 secundară a tabloului construit în urma executării secvenței de mai sus.  
a. 6      b. 12      c. 18      d. 30
- Utilizând metoda backtracking se generează toate posibilitățile de a forma șiraguri din câte 3 mărgelile de culori distincte din mulțimea {**roșu, galben, verde, albastru, violet**}. Două șiraguri sunt distincte dacă diferă prin cel puțin o culoare a mărgelilor sau prin ordinea acestora. Primele patru soluții generate sunt, în această ordine: (**roșu, galben, verde**), (**roșu, galben, albastru**), (**roșu, galben, violet**), (**roșu, verde, galben**). Indicați penultima soluție generată.  
a. (**violet, albastru, galben**)      b. (**violet, verde, albastru**)  
c. (**verde, albastru, galben**)      d. (**verde, violet, albastru**)
- Un arbore cu 10 noduri, numerotate de la 1 la 10, este reprezentat prin vectorul de „tați” (2, 5, 4, 0, 4, 1, 5, 5, 3, 4). Indicați numărul de descendenți direcți („fii”) ai nodului 5.  
a. 1      b. 2      c. 3      d. 4
- Un graf orientat cu 6 vârfuri, numerotate de la 1 la 6, este reprezentat prin matricea de adiacență ale cărei prime 5 linii sunt scrise alăturat. Două drumuri sunt distincte dacă diferă prin cel puțin un arc traversat. Indicați gradul exterior maxim al vârfului 6, astfel încât să existe doar trei drumuri elementare cu extremitatea inițială în vârful 2 și extremitatea finală în vârful 4.  

```
0 0 0 1 0 0
1 0 0 0 1 1
0 0 0 1 0 0
0 0 0 0 0 0
0 0 1 0 0 1
.....
```

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  $a \div b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$  și cu  $\lfloor c \rfloor$  partea întreagă a numărului real  $c$ .
- a. Scrieți ce se afișează dacă se citește, în această ordine, numerele 8 și 3. **(6p.)**
- b. Dacă pentru variabila  $k$  se citește valoarea 5, scrieți cea mai mică și cea mai mare valoare care pot fi citite pentru variabila  $n$  astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, ultimul număr afișat să fie 10. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat ultima structură **pentru...execută** cu o structură repetitivă de alt tip. **(6p.)**
2. Subprogramul  $f$  este definit alăturat. Scrieți ce valori au  $f(0)$ , respectiv  $f(2020)$ . **(6p.)**
3. Un graf neorientat cu 9 noduri, numerotate de la 1 la 9, are muchiile  $[1, 3]$ ,  $[1, 5]$ ,  $[2, 6]$ ,  $[2, 8]$ ,  $[2, 9]$ ,  $[3, 5]$ ,  $[4, 7]$ ,  $[6, 9]$ ,  $[8, 9]$ . Scrieți numărul componentelor conexe ale grafului și enumerați nodurile care fac parte din componenta conexă cu cele mai puține noduri. **(6p.)**

```

citește n,k
(numere naturale nenule)
t←1
pentru i←1,⌊n/k⌋ execută
    pentru j←1,k execută
        scrie t, ' '
    t←t+1
pentru i←n%k,1,-1 execută
    scrie t, ' '

```

```

int f (int n)
{ if (n!=0)
    return n%20+f(n/20);
  return 0;
}

```

**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 impare distincte din scrierea acestuia.
- Scrieți definiția completă a subprogramului.  
**Exemplu:** dacă  $n=4713835$ , după apel  $s=16$  ( $16=7+1+3+5$ ), iar dacă  $n=48$ , după apel  $s=0$ . **(10p.)**
2. Un cuvânt este **prefix** al unui alt cuvânt dacă se obține din acesta, prin eliminarea ultimelor sale litere. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $n \in [2, 20]$ ) 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 prefix pe ultimul cuvânt citit. Dacă nu există astfel de cuvinte, se afișează pe ecran mesajul **nu exista**.  
**Exemplu:** dacă  $n=6$  și se citește cuvintele alăturate, pe ecran se afișează  
**raita raid raion** **(10p.)**
3. Numim **k-secvență** într-un șir de numere naturale, o succesiune de termeni aflați pe poziții consecutive în șir, cu proprietatea că sunt divizibili cu numărul natural nenul  $k$ . **Lungimea** secvenței este egală cu numărul de termeni ai săi. Fișierul **bac.txt** conține numere naturale din intervalul  $[0, 10^9]$ : pe prima linie un număr nenul  $k$ , iar pe a doua linie un șir de cel mult  $10^6$  numere, separate prin câte un spațiu. Cel puțin un termen din șir este divizibil cu  $k$ . Se cere să se afișeze pe ecran două valori, separate printr-un spațiu, reprezentând lungimea maximă a unei  $k$ -secvențe din șirul aflat în fișier, respectiv numărul de astfel de secvențe. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.  
**Exemplu:** dacă fișierul are conținutul alăturat, se afișează **4 2**
- a. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
- b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**