

**Examenul de bacalaureat național 2020**  
**Proba E. d)**  
**Informatică**  
**Limbajul Pascal**

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 Pascal cu valoarea `true` dacă  $x$  memorează un divizor al lui 2020.
  - `2020 div (2020 div x)=0`
  - `2020 div (2020 mod x)=0`
  - `2020 mod (2020 mod x)=0`
  - `2020 mod (2020 div 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 to 5 do
  for j:=0 to 5 do
    a[i,j]:= (i mod 2)*j+(j mod 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)**

- |  |  |
|--|--|
| <p>1. Algoritmul alăturat este reprezentat în pseudocod.</p> <p>a. Scrieți ce se afișează dacă se citesc, în această ordine, numerele 2 și 3. <b>(6p.)</b></p> <p>b. Scrieți două seturi de valori din intervalul <math>[1, 10^2]</math> care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, ultima valoare afișată să fie 20. <b>(6p.)</b></p> <p>c. Scrieți programul Pascal corespunzător algoritmului dat. <b>(10p.)</b></p> <p>d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat una dintre structurile <b>pentru...execută</b> cu o structură repetitivă de alt tip. <b>(6p.)</b></p> | <pre> 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   ■ </pre> |
| <p>2. Subprogramul <math>f</math> este definit alăturat. Scrieți ce valori au <math>f(0)</math>, respectiv <math>f(23575209)</math>. <b>(6p.)</b></p>  | <pre> function f(n: longint): longint; begin if n &lt;&gt; 0 then f := n mod 2 + f(n div 100)       else f := 0 end; </pre>  |
| <p>3. Un graf neorientat cu 9 noduri, numerotate de la 1 la 9, are muchiile <math>[1, 2]</math>, <math>[1, 6]</math>, <math>[2, 6]</math>, <math>[3, 7]</math>, <math>[4, 5]</math>, <math>[4, 9]</math>. Scrieți numărul componentelor conexe ale grafului și câte dintre acestea au un număr maxim de noduri. <b>(6p.)</b></p>   |  |

**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 Pascal 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      | isihast      |
| paratirisi   | paratirisi   |
| isi          | isi          |
| meremetisire | meremetisire |
| acolisitor   | acolisitor   |
| hiritisi     | hiritisi     |
| paraponisit  | 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**
- |   |   |     |     |    |   |   |     |      |    |      |    |    |
|---|---|-----|-----|----|---|---|-----|------|----|------|----|----|
| 7 | 3 | 200 | 100 | 10 | 9 | 6 | 100 | 1000 | 40 | 1002 | 20 | 30 |
|---|---|-----|-----|----|---|---|-----|------|----|------|----|----|
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
- b. Scrieți programul Pascal corespunzător algoritmului proiectat. **(8p.)**