

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $[c]$ partea întreagă a numărului real c .

- Scriveți valoarea afișată dacă se citește, în această ordine, numerele 5, 15, 27, 10, 1, 17. **(6p.)**
- Dacă pentru n se citește valoarea 2, scrieți un set de numere distincte din intervalul $[0, 10^3]$ care pot fi citite în continuare, astfel încât, în urma executării algoritmului, să se afișeze valoarea 4. **(6p.)**
- Scriveți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- Scriveți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura **pentru...execută** cu o structură repetitivă de tip **cât timp...execută**. **(6p.)**

```

citește n
    (număr natural nenul)
p ← 1
pentru i ← 1, n execută
    citește x
    (număr natural)
    repetă
        x ← [x/3]
    până când x ≤ 3
    dacă x ≠ 0 atunci
        p ← p * x
scrie p
    
```

- Un arbore cu 8 noduri, numerotate de la 1 la 8, este reprezentat prin vectorul de „tați”: (3,0,2,5,2,5,1,5). Enumerați, în ordinea parcurgerii lor, nodurile celui mai lung lanț elementar care are extremitatea inițială în rădăcină. **(6p.)**

- Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 9 linii și 9 coloane, numerotate începând de la 0, având inițial toate elementele nule. Scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie cu instrucțiuni adecvate, dintre care cel mult patru de atribuire, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat.

```

for (i=0; i<9; i++)
    for (j=0; j<9; j++)
        .....
    
```

(6p.)

4	4	4	4	2	2	2	2	2
4	4	4	2	2	2	2	2	2
4	4	2	2	2	2	2	2	2
4	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	4
2	2	2	2	2	2	2	4	4
2	2	2	2	2	2	4	4	4
2	2	2	2	2	4	4	4	4

SUBIECTUL al III-lea

(30 de puncte)

- Un număr natural se numește **major impar** dacă suma divizorilor săi proprii impari este strict mai mare decât suma divizorilor săi proprii pari. Divizorii proprii ai unui număr sunt divizorii săi naturali diferiți de 1 și de el însuși.

Exemplu: 18 este număr major impar (divizorii săi proprii pari sunt 2, 6, cei impari 3, 9, iar $3+9>2+6$).

Subprogramul **majImp** are doi parametri, a și b , prin care primește câte un număr natural ($2 \leq a \leq b \leq 10^4$).

Subprogramul returnează cel mai mic număr major impar din intervalul $[a, b]$, sau valoarea 0, dacă în interval nu există un astfel de număr. Scrieți în C/C++ definiția completă a subprogramului.

Exemplu: dacă $a=16$, $b=30$, atunci subprogramul returnează 18. **(10p.)**

- Într-un text, de cel mult 100 de caractere, cuvintele sunt formate din litere ale alfabetului englez și sunt separate prin câte un spațiu. Textul are cel puțin două cuvinte.

Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat mai sus și afișează pe ecran mesajul **DA** și un număr natural n , separate printr-un spațiu, dacă toate cuvintele din text au câte n litere, sau mesajul **NU** în cazul în care nu toate cuvintele au același număr de litere.

Exemplu: dacă textul citit este **Ana are cel mai bun mar** se afișează pe ecran **DA 3**

iar dacă textul citit este **Ana are cel mai dulce mar** se afișează pe ecran **NU** **(10p.)**

- De-a lungul unui traseu montan este utilizată o succesiune de marcaje turistice, care trebuie urmate în acea ordine. Pentru fiecare marcaj se cunoaște cota (înălțimea, măsurată în metri) la care este plasat. Numim **scară** într-un traseu o secvență de marcaje aflate pe poziții consecutive în cadrul traseului, care au drept cote numere consecutive, ordonate strict crescător. O scară este formată din cel puțin două marcaje, iar lungimea acesteia este egală cu numărul de marcaje care o compun.

Fișierul **bac.txt** conține un șir de cel mult 10^6 numere naturale din intervalul $[10, 10^4]$, separate prin câte un spațiu, reprezentând cotele marcajelor turistice din cadrul unui traseu, în ordinea în care se succed în acesta. Se cere să se afișeze pe ecran, separate prin câte un spațiu, în ordine strict crescătoare, cotele corespunzătoare marcajelor unei scări de lungime maximă pe acest traseu. Dacă în cadrul traseului există mai multe astfel de scări, se afișează doar cotele corespunzătoare marcajelor uneia dintre ele, iar dacă nu există nicio scară, pe ecran se afișează mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu: dacă fișierul conține numerele 500 600 601 405 569 570 700 701 625 626 627 520 atunci pe ecran se afișează 625 626 627

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

- Scriveți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

Examenul național de bacalaureat 2024

Proba E. d)
INFORMATICĂ
Limbajul Pascal

Varianta 3

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 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.

- Indicați expresia Pascal care are valoarea **true** dacă și numai dacă numerele memorate în variabilele întregi **x** și **y** sunt pare.
 - (x mod 2=0) and ((y+1) mod 2<>0)**
 - (x-y) mod 2=0**
 - (x+y) mod 2=0**
 - x mod 2=y mod 2**
- Subprogramul **f** este definit alăturat.
Indicați ce se afișează în urma apelului de mai jos.
f(2020, 0);

```
procedure f(x,y:longint);  
begin if x<10 then write(x)  
      else  
        begin f(x div 10,y+1);  
              write(x mod 10)  
        end;  
      write(y)  
end;
```

 - 23020**
 - 2022100**
 - 02023210**
 - 23022100**
- Utilizând metoda backtracking se generează toate permutările elementelor mulțimii ordonate astfel: {**1, 2, 3, 4, 5, 6**}; pentru fiecare permutare, pe primele trei poziții sunt doar valori pare, iar pe ultimele trei poziții sunt doar valori impare. Primele șase permutări generate sunt, în această ordine: (**2,4,6,1,3,5**), (**2,4,6,1,5,3**), (**2,4,6,3,1,5**), (**2,4,6,3,5,1**), (**2,4,6,5,1,3**), (**2,4,6,5,3,1**). Indicați a șaptea permutare generată.
 - (4,2,6,1,5,3)**
 - (4,2,6,1,3,5)**
 - (2,6,4,1,3,5)**
 - (2,4,6,5,3,2)**
- Variabila **x** memorează, pentru fiecare dintre cele 20 de sortimente de ciocolată, următoarele date: tipul (litera **N** pentru ciocolată neagră și litera **L** pentru ciocolată cu lapte) și prețul produsului. Indicați o expresie a cărei valoare este egală cu tipul celui de al 10-lea sortiment de ciocolată.

```
type ciocolata=record  
      tip:char;  
      pret:real  
end;  
var x:array[1..20] of ciocolata;
```

 - x.ciocolata[10].tip**
 - x.tip[10]**
 - x[10].ciocolata.tip**
 - x[10].tip**
- Într-un graf neorientat, cu 10 muchii, două noduri au gradul 0, șase noduri au grade impare, iar celelalte noduri au grade pare, nenule. Indicați numărul maxim de noduri ale grafului.
 - 17**
 - 15**
 - 12**
 - 10**

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $[c]$ partea întreagă a numărului real c .

- Scriveți valoarea afișată dacă se citește, în această ordine, numerele 5, 15, 27, 10, 1, 17. **(6p.)**
- Dacă pentru n se citește valoarea 2, scrieți un set de numere distincte din intervalul $[0, 10^3]$ care pot fi citite în continuare, astfel încât, în urma executării algoritmului, să se afișeze valoarea 4. **(6p.)**
- Scriveți programul Pascal corespunzător algoritmului dat. **(10p.)**
- Scriveți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura **pentru...execută** cu o structură repetitivă de tip **cât timp...execută**. **(6p.)**

```

citește n
    (număr natural nenul)
p ← 1
pentru i ← 1, n execută
    citește x
    (număr natural)
    repetă
        x ← [x/3]
    până când x ≤ 3
    dacă x ≠ 0 atunci
        p ← p * x
    ■
scrie p
    
```

- Un arbore cu 8 noduri, numerotate de la 1 la 8, este reprezentat prin vectorul de „tați”: (3,0,2,5,2,5,1,5). Enumerați, în ordinea parcurgerii lor, nodurile celui mai lung lanț elementar care are extremitatea inițială în rădăcină. **(6p.)**

- Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 9 linii și 9 coloane, numerotate începând de la 0, având inițial toate elementele nule. Scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie cu instrucțiuni adecvate, dintre care cel mult patru de atribuire, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat.

```

for i:=0 to 8 do
    for j:=0 to 8 do
        .....
    
```

(6p.)

4	4	4	4	2	2	2	2	2
4	4	4	2	2	2	2	2	2
4	4	2	2	2	2	2	2	2
4	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	4
2	2	2	2	2	2	2	4	4
2	2	2	2	2	2	4	4	4
2	2	2	2	2	4	4	4	4

SUBIECTUL al III-lea

(30 de puncte)

- Un număr natural se numește **major impar** dacă suma divizorilor săi proprii impari este strict mai mare decât suma divizorilor săi proprii pari. Divizorii proprii ai unui număr sunt divizorii săi naturali diferiți de 1 și de el însuși.

Exemplu: 18 este număr major impar (divizorii săi proprii pari sunt 2, 6, cei impari 3, 9, iar $3+9>2+6$).

Subprogramul **majImp** are doi parametri, a și b , prin care primește câte un număr natural ($2 \leq a \leq b \leq 10^4$).

Subprogramul returnează cel mai mic număr major impar din intervalul $[a, b]$, sau valoarea 0, dacă în interval nu există un astfel de număr. Scrieți în Pascal definiția completă a subprogramului.

Exemplu: dacă $a=16$, $b=30$, atunci subprogramul returnează 18. **(10p.)**

- Într-un text, de cel mult 100 de caractere, cuvintele sunt formate din litere ale alfabetului englez și sunt separate prin câte un spațiu. Textul are cel puțin două cuvinte.

Scrieți un program Pascal care citește de la tastatură un text de tipul precizat mai sus și afișează pe ecran mesajul **DA** și un număr natural n , separate printr-un spațiu, dacă toate cuvintele din text au câte n litere, sau mesajul **NU** în cazul în care nu toate cuvintele au același număr de litere.

Exemplu: dacă textul citit este **Ana are cel mai bun mar** se afișează pe ecran **DA 3**

iar dacă textul citit este **Ana are cel mai dulce mar** se afișează pe ecran **NU** **(10p.)**

- De-a lungul unui traseu montan este utilizată o succesiune de marcaje turistice, care trebuie urmate în acea ordine. Pentru fiecare marcaj se cunoaște cota (înălțimea, măsurată în metri) la care este plasat. Numim **scară** într-un traseu o secvență de marcaje aflate pe poziții consecutive în cadrul traseului, care au drept cote numere consecutive, ordonate strict crescător. O scară este formată din cel puțin două marcaje, iar lungimea acesteia este egală cu numărul de marcaje care o compun.

Fișierul **bac.txt** conține un șir de cel mult 10^6 numere naturale din intervalul $[10, 10^4]$, separate prin câte un spațiu, reprezentând cotele marcajelor turistice din cadrul unui traseu, în ordinea în care se succed în acesta. Se cere să se afișeze pe ecran, separate prin câte un spațiu, în ordine strict crescătoare, cotele corespunzătoare marcajelor unei scări de lungime maximă pe acest traseu. Dacă în cadrul traseului există mai multe astfel de scări, se afișează doar cotele corespunzătoare marcajelor uneia dintre ele, iar dacă nu există nicio scară, pe ecran se afișează mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu: dacă fișierul conține numerele 500 600 601 405 569 570 700 701 625 626 627 520 atunci pe ecran se afișează **625 626 627**

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

- Scriveți programul Pascal corespunzător algoritmului proiectat. **(8p.)**

Examenul național de bacalaureat 2024
Proba E. d)
INFORMATICĂ

BAREM DE EVALUARE ȘI DE NOTARE
(comun pentru limbajele C/C++ și Pascal)

Varianța 3

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

- Se punctează orice modalitate de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit în barem. Nu se acordă fracțiuni de punct.
- Se acordă zece puncte din oficiu. Nota finală se calculează prin împărțirea la zece a punctajului total acordat pentru lucrare.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț (de exemplu tipuri întregi cu semn pentru memorarea numerelor naturale, dimensiune a tablourilor) este acceptată din punctul de vedere al corectitudinii programului, dacă acest lucru nu afectează funcționarea sa.

SUBIECTUL I (20 de puncte)

1a 2d 3c 4d 5b	5x4p.
----------------	-------

SUBIECTUL al II - lea (40 de puncte)

1.	a. Răspuns corect: 9	6p.	
	b. Pentru răspuns corect	6p.	Se acordă câte 2p. pentru fiecare aspect specific (număr de valori scrise, prima valoare, a doua valoare) conform cerinței. Se punctează valori naturale din reuniunea intervalelor de forma $[2 \cdot 3^k, 3^{k+1})$, cu k număr natural, $k \in \{1, 2, 3, 4, 5\}$.
	c. Pentru program corect -declaraire a variabilelor -citire a datelor -afișare a datelor -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d. Pentru algoritm pseudocod corect -utilizare a unei structuri repetitive de tipul indicat (*) -aspecte specifice ale secvenței obținute prin înlocuire, conform cerinței (**) -algoritm complet, corectitudine globală a algoritmului ¹⁾	6p. 2p. 3p. 1p.	(*) Se acordă punctajul pentru orice formă de structură repetitivă de tipul cerut (cât timp ... execută, while ... do etc.). (**) Se acordă câte 1p. pentru fiecare aspect specific (inițializare a contorului, expresie de continuare, actualizare a contorului) conform cerinței.
2.	Răspuns corect: 2, 3, 1, 7	6p.	Se acordă câte 2p. pentru fiecare aspect specific (lanț cu o extremitate în rădăcină, lanț elementar, lungime maximă a lanțului) conform cerinței.
3.	Pentru rezolvare corectă -expresie de accesare a unui element al tabloului -valori ale elementelor tabloului atribuite conform cerinței (*) -corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect specific (identificare a cel puțin unei relații între valoarea elementului și poziția acestuia în tablou, valori suport în corelare cu pozițiile elementelor utilizând numărul indicat de instrucțiuni de atribuire) conform cerinței.

SUBIECTUL al III - lea

(30 de puncte)

1. Pentru subprogram corect -antet al subprogramului (*) -determinare a valorii cerute (**) -instrucțiune/instrucțiuni de returnare a rezultatului și tratare a cazului 0 -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului¹⁾	10p. (*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri de intrare) conform cerinței. 2p. 6p. (**) Se acordă câte 1p. pentru fiecare aspect specific (identificare a unui divizor par/impar, algoritm de bază pentru suma unei serii de valori, divizori suport pentru fiecare sumă, identificare a unui număr major impar, algoritm de bază pentru identificarea primului/celui mai mic număr cu o anumită proprietate dintr-o serie, numere suport verificate pentru determinarea numărului major impar cerut) conform cerinței. 1p. 1p.
2. Pentru program corect -declarare a unei variabile care să permită memorarea unui text, conform cerinței -citire a datelor -verificare a proprietății cerute (*) -afișare a datelor în formatul cerut -declarare a variabilelor simple, corectitudine globală a programului¹⁾	10p. (*) Se acordă câte 2p. pentru fiecare aspect specific (algoritm principal corect de verificare a unei proprietăți, determinare a lungimii unui cuvânt, cuvinte suport verificate) conform cerinței. 1p. 1p. 6p. 1p. 1p.
3. a. Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență b. Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorilor cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor și tratare a cazului nu exista, corectitudine globală a programului¹⁾	2p. (*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. 1p. 1p. 8p. (**) Se acordă numai 3p. dacă algoritmul este principal corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. 1p. 5p. 1p. 1p. 1p. (***) Se acordă punctajul numai pentru un algoritm liniar, care utilizează eficient memoria. O soluție posibilă parcurge o dată fișierul, memorând pentru scara curentă și pentru cea căutată lungimea (lgCrt, respectiv lgMax, inițial egale cu 0) și ultimul termen (ultCrt, inițializat cu 0, respectiv ultMax) ale acestora; dacă numărul x citit din fișier la pasul curent face parte din scara curentă ($x = \text{ultCrt} + 1$) se actualizează lgCrt, altfel se actualizează datele privind o nouă scară, deci $\text{lgCrt} = 1$. La fiecare pas se actualizează $\text{ultCrt} = x$ și, după caz, lgMax și ultMax. Se afișează lgMax numere consecutive, primul dintre acestea fiind $\text{ultMax} - \text{lgMax} + 1$. Mesajul se afișează dacă $\text{lgMax} < 2$.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.