

Examenul național de bacalaureat 2023  
Proba E. d)  
Informatică  
Limbajul Pascal

Varianta 5

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 cea mai mare valoare, comparativ cu celelalte trei expresii.  
a.  $20 * 23 \text{ div } (2 * 2)$     b.  $20 \text{ div } 2 * 23 \text{ div } 2$     c.  $(20 * 23) \text{ div } 2$     d.  $(20 * 23) \text{ div } 2 * 2$
- Subprogramul **f** este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.  
`f (23);`  
a. 100111    b. 111010    c. 010111    d. 01251123  

```
procedure f(n:integer);  
begin if n<>0 then f(n div 2);  
       write(n mod 2)  
end;
```
- Variabila **k** este de tip întreg, iar variabila **s** permite memorarea unui șir de maximum 50 de caractere. Indicați valoarea variabilei **k** în urma executării secvenței alăturate.  
a. 7    b. 6    c. 2    d. 1  

```
s:='bac2023';  
delete(s,ord(s[3])-ord('a')+1,7);  
k:=length(s);
```
- Indicați un vector de „tați” corespunzător unui arbore cu 7 noduri, în care cel puțin unul dintre noduri are trei ascendenți.  
a. 0, 1, 2, 1, 1, 1, 2    b. 3, 0, 2, 1, 3, 2, 1    c. 4, 3, 0, 3, 4, 4, 3    d. 5, 4, 3, 0, 2, 3, 4
- O companie de colectare a fructelor are 6 depozite, numerotate de la 1 la 6: depozitele 1, 3 și 5 conțin mere, depozitele 2 și 4 conțin pere, iar depozitul 6 conține piersici. Compania a construit 4 benzi de transport unidirecțional: de la depozitul 1 la depozitul 5, de la depozitul 5 la depozitul 2, de la depozitul 5 la depozitul 4 și de la depozitul 6 la depozitul 1. Dacă depozitele reprezintă vârfurile unui graf orientat, iar benzile de transport reprezintă arcele acestuia, indicați numărul minim de benzi de transport care pot fi adăugate, astfel încât graful obținut să aibă trei componente tare conexe, fiecare dintre acestea având vârfuri care corespund depozitelor cu același tip de fructe.  
a. 4    b. 3    c. 2    d. 1

**SUBIECTUL al II-lea** (40 de puncte)

**1. Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu **a**%**b** restul împărțirii numărului natural **a** la numărul natural nenul **b**, și cu **[c]** partea întregă a numărului real **c**.

- Scrieți valoarea afișată dacă se citește numărul **6907512**. (6p.)
- Scrieți două valori distincte din intervalul **[100, 999]** care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze o valoare identică cu cea citită. (6p.)
- Scrieți programul Pascal corespunzător algoritmului dat. (10p.)
- Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **cât timp...execută** cu o structură repetitivă cu test final. (6p.)

```
citește x (număr natural)  
p ← 1; m ← -1  
cât timp p ≤ x execută  
| c ← [x/p] % 10  
| dacă c > m atunci  
|   m ← c; p ← p * 10  
| altfel  
|   x ← [x / (p * 10)] * p + x % p  
|  
|  
dacă m ≥ 0 atunci scrie x  
altfel scrie "nul"
```

2. Utilizând metoda backtracking, se generează toate amestecurile de apă provenită din surse distincte din mulțimea {lac, mare, ocean, ploaie, râu}, astfel încât o sursă să fie de apă sărată și una sau două surse să fie de apă dulce. Marea și oceanul sunt surse de apă sărată, iar lacul, ploaia și râul sunt surse de apă dulce. Două amestecuri sunt distincte dacă diferă prin cel puțin o sursă a apei. Primele patru soluții obținute sunt, în această ordine: (lac, mare), (lac, mare, ploaie), (lac, mare, râu) și (lac, ocean). Scrieți soluția generată imediat înainte și soluția generată imediat după (ocean, ploaie). (6p.)
3. Variabila  $f$  memorează, pentru fiecare dintre cele 10 soiuri de lalele care se vând într-o florărie, caracteristicile acestora: denumirea (șir de maximum 20 de caractere) și stocul, exprimat prin numărul de fire și prețul unui fir, în lei (numere naturale). Știind că expresiile de mai jos au ca valori denumirea primului soi de lalele, respectiv suma, în lei, necesară pentru a cumpăra toate lalelele din acest soi, scrieți în limbajul Pascal definiția unui tip de date cu numele `lalea`, înregistrare care să permită memorarea informațiilor menționate pentru un soi de lalea, și declarați corespunzător variabila  $f$ .  
`f[0].denumire      f[0].stoc.nrFire*f[0].stoc.pretFir` (6p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Un număr natural nenul,  $n$ , se numește număr **abundent** dacă  $s(n)/n > s(k)/k$ , pentru orice număr natural nenul  $k$  ( $k \leq n-1$ ), unde s-a notat cu  $s(i)$  suma divizorilor pozitivi ai numărului natural nenul  $i$ . Subprogramul **abundent** are un singur parametru,  $n$ , prin care primește un număr natural ( $n \in [2, 10^6]$ ). Subprogramul returnează valoarea 1, dacă  $n$  este un număr abundent, sau valoarea 0, în caz contrar. Scrieți definiția completă a subprogramului.  
**Exemplu:** pentru  $n=6$ , subprogramul returnează valoarea 1 ( $s(6)/6=2$ , iar cel mai mare raport obținut pentru valori strict mai mici decât 6 este  $s(4)/4=1.75$ ), iar pentru  $n=7$  sau  $n=8$ , subprogramul returnează valoarea 0 ( $s(7)/7=1.14$ ,  $s(8)/8=1.87$ ). (10p.)
2. Pentru a identifica punctele în care se concentrează apa în albia unui râu în cazul secetei, se determină talvegul acesteia – linia care unește punctele cele mai adânci ale albiei. În acest scop s-au stabilit  $ns$  secțiuni transversale pe cursul apei, numerotate începând de la 1, și în cadrul fiecărei secțiuni s-a măsurat adâncimea apei în  $np$  puncte, numerotate începând de la 1. Din fiecare secțiune, în ordine, se include în talveg cel mai adânc punct al acesteia, iar dacă în secțiune sunt mai multe puncte aflate la aceeași adâncime, maximă, se va lua în considerare doar primul dintre ele, ca în exemplu. Scrieți un program Pascal care citește de la tastatură două numere naturale,  $ns$  și  $np$  ( $ns \in [1, 10^3]$ ,  $np \in [1, 50]$ ), și cele  $ns \cdot np$  elemente ale unui tablou bidimensional, valori naturale din intervalul  $[0, 10^2]$ . Fiecare linie a tabloului corespunde câte unei secțiuni, în ordinea numerotării acestora, iar valorile memorate pe linie reprezintă adâncimile celor  $np$  puncte stabilite pentru acea secțiune, în ordinea numerotării lor. Programul afișează pe ecran, pentru fiecare secțiune, o pereche formată din numărul de ordine al secțiunii și numărul de ordine al punctului său care s-a inclus în talveg. Numerele din fiecare pereche sunt afișate separate prin câte un caracter : (două puncte), iar fiecare pereche este urmată de un spațiu.  
**Exemplu:** pentru  $ns=6$ ,  $np=4$  și tabloul alăturat, se afișează pe ecran valorile:  
1:3 2:2 3:2 4:2 5:4 6:3 (10p.)
- |   |   |   |   |
|---|---|---|---|
| 2 | 4 | 5 | 3 |
| 2 | 6 | 6 | 3 |
| 1 | 5 | 2 | 5 |
| 1 | 3 | 3 | 3 |
| 3 | 4 | 3 | 5 |
| 0 | 1 | 2 | 1 |
3. Un număr natural  $x$  este numit prefix al unui număr natural  $y$  dacă se obține din acesta prin eliminarea a cel puțin unei cifre de la dreapta sa, și este numit sufix al lui  $y$  dacă se obține din acesta prin eliminarea a cel puțin unei cifre de la stânga sa.  
**Exemplu:** 15 este prefix pentru 154 sau 1521, este sufix pentru 3415 sau 5115, dar nu este nici prefix, nici sufix pentru 15.  
Fișierul `bac.txt` conține maximum  $10^6$  numere naturale din intervalul  $[10, 10^4]$ , separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul valorilor de două cifre care apar de același număr de ori ca sufix, respectiv ca prefix al numerelor din șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.  
**Exemplu:** dacă fișierul are conținutul  
342 1684 2134 5434 111 98 98 3405 3412 7016 8634 1010 102 310  
se afișează pe ecran: 4 (pentru valorile 10, 11, 16, 34).  
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
b. Scrieți programul Pascal corespunzător algoritmului proiectat. (8p.)

Examenul național de bacalaureat 2023  
Proba E. d)  
Informatică  
Limbajul C/C++

Varianta 5

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 C/C++ care are cea mai mare valoare, comparativ cu celelalte trei expresii.  
a.  $20 * 23 / (2 * 2)$       b.  $20 / 2 * 23 / 2$       c.  $(20 * 23) / 2$       d.  $(20 * 23) / 2 * 2$
- Subprogramul **f** este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.  
`f(23);`  
a. 100111      b. 111010      c. 010111      d. 01251123  

```
void f (int n)
{ if (n!=0) f(n/2);
  cout<<n%2; | printf ("%d", n%2);
}
```
- Variabila **k** este de tip întreg, iar variabila **s** permite memorarea unui șir de maximum 50 de caractere. Indicați valoarea variabilei **k** în urma executării secvenței alăturate.  
a. 7      b. 6      c. 2      d. 1  

```
strcpy (s, "bac2023");
s[s[2]-'a'] = '\0';
k=strlen(s);
```
- Indicați un vector de „tați” corespunzător unui arbore cu 7 noduri, în care cel puțin unul dintre noduri are trei ascendenți.  
a. 0, 1, 2, 1, 1, 1, 2      b. 3, 0, 2, 1, 3, 2, 1      c. 4, 3, 0, 3, 4, 4, 3      d. 5, 4, 3, 0, 2, 3, 4
- O companie de colectare a fructelor are 6 depozite, numerotate de la 1 la 6: depozitele 1, 3 și 5 conțin mere, depozitele 2 și 4 conțin pere, iar depozitul 6 conține piersici. Compania a construit 4 benzi de transport unidirecțional: de la depozitul 1 la depozitul 5, de la depozitul 5 la depozitul 2, de la depozitul 5 la depozitul 4 și de la depozitul 6 la depozitul 1. Dacă depozitele reprezintă vârfurile unui graf orientat, iar benzile de transport reprezintă arcele acestuia, indicați numărul minim de benzi de transport care pot fi adăugate, astfel încât graful obținut să aibă trei componente tare conexe, fiecare dintre acestea având vârfuri care corespund depozitelor cu același tip de fructe.  
a. 4      b. 3      c. 2      d. 1

**SUBIECTUL al II-lea** (40 de puncte)

**1. Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu **a**%**b** restul împărțirii numărului natural **a** la numărul natural nenul **b**, și cu **[c]** partea întregă a numărului real **c**.

- Scrieți valoarea afișată dacă se citește numărul 6907512. (6p.)
- Scrieți două valori distincte din intervalul [100, 999] care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze o valoare identică cu cea citită. (6p.)
- Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **cât timp...execută** cu o structură repetitivă cu test final. (6p.)

```
citește x (număr natural)
p ← 1; m ← -1
cât timp p ≤ x execută
| c ← [x/p] % 10
| dacă c > m atunci
| | m ← c; p ← p * 10
| altfel
| | x ← [x / (p * 10)] * p + x % p
| ■
| ■
dacă m ≥ 0 atunci scrie x
| altfel scrie "nul"
| ■
```

2. Utilizând metoda backtracking, se generează toate amestecurile de apă provenită din surse distincte din mulțimea {lac, mare, ocean, ploaie, râu}, astfel încât o sursă să fie de apă sărată și una sau două surse să fie de apă dulce. Marea și oceanul sunt surse de apă sărată, iar lacul, ploaia și râul sunt surse de apă dulce. Două amestecuri sunt distincte dacă diferă prin cel puțin o sursă a apei. Primele patru soluții obținute sunt, în această ordine: (lac, mare), (lac, mare, ploaie), (lac, mare, râu) și (lac, ocean). Scrieți soluția generată imediat înainte și soluția generată imediat după (ocean, ploaie). (6p.)
3. Variabila  $f$  memorează, pentru fiecare dintre cele 10 soiuri de lalele care se vând într-o florărie, caracteristicile acestora: denumirea (șir de maximum 20 de caractere) și stocul, exprimat prin numărul de fire și prețul unui fir, în lei (numere naturale). Știind că expresiile de mai jos au ca valori denumirea primului soi de lalele, respectiv suma, în lei, necesară pentru a cumpăra toate lalelele din acest soi, scrieți în limbajul C/C++ definiția unei structuri cu eticheta `lalea`, care să permită memorarea informațiilor menționate pentru un soi de lalea, și declarați corespunzător variabila  $f$ .  
`f[0].denumire      f[0].stoc.nrFire*f[0].stoc.pretFir` (6p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Un număr natural nenul,  $n$ , se numește număr **abundent** dacă  $s(n)/n > s(k)/k$ , pentru orice număr natural nenul  $k$  ( $k \leq n-1$ ), unde s-a notat cu  $s(i)$  suma divizorilor pozitivi ai numărului natural nenul  $i$ . Subprogramul `abundent` are un singur parametru,  $n$ , prin care primește un număr natural ( $n \in [2, 10^6]$ ). Subprogramul returnează valoarea 1, dacă  $n$  este un număr abundent, sau valoarea 0, în caz contrar. Scrieți definiția completă a subprogramului.  
**Exemplu:** pentru  $n=6$ , subprogramul returnează valoarea 1 ( $s(6)/6=2$ , iar cel mai mare raport obținut pentru valori strict mai mici decât 6 este  $s(4)/4=1.75$ ), iar pentru  $n=7$  sau  $n=8$ , subprogramul returnează valoarea 0 ( $s(7)/7=1.14$ ,  $s(8)/8=1.87$ ). (10p.)
2. Pentru a identifica punctele în care se concentrează apa în albia unui râu în cazul secetei, se determină talvegul acesteia – linia care unește punctele cele mai adânci ale albiei. În acest scop s-au stabilit  $ns$  secțiuni transversale pe cursul apei, numerotate începând de la 1, și în cadrul fiecărei secțiuni s-a măsurat adâncimea apei în  $np$  puncte, numerotate începând de la 1. Din fiecare secțiune, în ordine, se include în talveg cel mai adânc punct al acesteia, iar dacă în secțiune sunt mai multe puncte aflate la aceeași adâncime, maximă, se va lua în considerare doar primul dintre ele, ca în exemplu. Scrieți un program C/C++ care citește de la tastatură două numere naturale,  $ns$  și  $np$  ( $ns \in [1, 10^3]$ ,  $np \in [1, 50]$ ), și cele  $ns \cdot np$  elemente ale unui tablou bidimensional, valori naturale din intervalul  $[0, 10^2]$ . Fiecare linie a tabloului corespunde câte unei secțiuni, în ordinea numerotării acestora, iar valorile memorate pe linie reprezintă adâncimile celor  $np$  puncte stabilite pentru acea secțiune, în ordinea numerotării lor. Programul afișează pe ecran, pentru fiecare secțiune, o pereche formată din numărul de ordine al secțiunii și numărul de ordine al punctului său care s-a inclus în talveg. Numerele din fiecare pereche sunt afișate separate prin câte un caracter : (două puncte), iar fiecare pereche este urmată de un spațiu.  
**Exemplu:** pentru  $ns=6$ ,  $np=4$  și tabloul alăturat, se afișează pe ecran valorile:  
`1:3 2:2 3:2 4:2 5:4 6:3` (10p.)
- |   |   |   |   |
|---|---|---|---|
| 2 | 4 | 5 | 3 |
| 2 | 6 | 6 | 3 |
| 1 | 5 | 2 | 5 |
| 1 | 3 | 3 | 3 |
| 3 | 4 | 3 | 5 |
| 0 | 1 | 2 | 1 |
3. Un număr natural  $x$  este numit prefix al unui număr natural  $y$  dacă se obține din acesta prin eliminarea a cel puțin unei cifre de la dreapta sa, și este numit sufix al lui  $y$  dacă se obține din acesta prin eliminarea a cel puțin unei cifre de la stânga sa.  
**Exemplu:** 15 este prefix pentru 154 sau 1521, este sufix pentru 3415 sau 5115, dar nu este nici prefix, nici sufix pentru 15.  
 Fișierul `bac.txt` conține maximum  $10^6$  numere naturale din intervalul  $[10, 10^4]$ , separate prin câte un spațiu. Se cere să se afișeze pe ecran numărul valorilor de două cifre care apar de același număr de ori ca sufix, respectiv ca prefix al numerelor din șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.  
**Exemplu:** dacă fișierul are conținutul  
`342 1684 2134 5434 111 98 98 3405 3412 7016 8634 1010 102 310`  
 se afișează pe ecran: 4 (pentru valorile 10, 11, 16, 34).  
**a.** Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
**b.** Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)

**Examenul național de bacalaureat 2023**  
**Proba E. d)**  
**Informatică**

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

**Varianta 5**

*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)

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

**SUBIECTUL al II - lea** (40 de puncte)

<b>1.</b>	<b>a. Răspuns corect: 9752</b>	<b>6p.</b>	
	<b>b. Pentru răspuns corect</b>	<b>6p.</b>	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței (oricare numere din intervalul cerut, cu cifre în ordine strict descrescătoare).
	<b>c. Pentru program corect</b> - variabile declarate, conform cerinței - date citite, conform cerinței - date afișate, conform cerinței - instrucțiune repetitivă, conform cerinței - instrucțiuni de decizie, conform cerinței (*) - atribuiri, conform cerinței - corectitudine globală a programului <sup>1)</sup>	<b>10p.</b> 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile de decizie este conform cerinței.
	<b>d. Pentru algoritm pseudocod corect</b> - structură repetitivă de tipul cerut (*) - aspecte specifice ale secvenței obținute prin înlocuire, conform cerinței (**) - algoritm complet, corectitudine globală a algoritmului <sup>1)</sup>	<b>6p.</b> 2p. 3p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul obținut nu este echivalent cu cel dat. Se va puncta orice formă de structură repetitivă conform cerinței (repetă...până când, repetă...cât timp, execută...cât timp, repeat ...until etc.). (**) Se acordă numai 2p. dacă doar unul dintre aspectele specifice (expresie logică pentru test final, echivalentă pentru cazul x=0) este conform cerinței.
<b>2.</b>	<b>Răspuns corect:</b> <b>(mare, râu)</b> <b>(ocean, ploaie, râu)</b>	<b>6p.</b>	Se acordă câte 3p. pentru fiecare soluție conform cerinței.
<b>3.</b>	<b>Pentru rezolvare corectă</b> - structură/înregistrare definită, conform cerinței (*) - variabilă declarată conform cerinței - corectitudine globală a secvenței <sup>1)</sup>	<b>6p.</b> 4p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (definiție principial corectă a unei structuri/înregistrări, câmpuri de pe primul nivel, câmpuri de pe al doilea nivel, etichetă/nume) conform cerinței.

**SUBIECTUL al III - lea**

**(30 de puncte)**

1.	<p><b>Pentru subprogram corect</b> - antet al subprogramului, conform cerinței (*) - proprietate verificată, conform cerinței (**) - instrucțiune/instrucțiuni de returnare a rezultatului, conform cerinței - variabile locale declarate conform cerinței, corectitudine globală a subprogramului<sup>1)</sup></p>	<p><b>10p.</b> 2p. 6p. 1p. 1p.</p>	<p>(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametru de intrare) conform cerinței. (**) Se acordă câte 1p. pentru fiecare aspect specific (identificare a unui divizor al unui număr, algoritm de bază pentru calculul sumei unei serii de valori, divizori suport însumați, împărțire reală pentru raport, algoritm principal corect de verificare a unei proprietăți, numere suport verificate) conform cerinței.</p>
2.	<p><b>Pentru program corect</b> - variabilă de tip tablou bidimensional, declarată conform cerinței - date citite conform cerinței - valori cu proprietatea cerută determinate (*) - date afișate în format conform cerinței - variabile simple declarate conform cerinței, corectitudine globală a programului<sup>1)</sup></p>	<p><b>10p.</b> 1p. 1p. 6p. 1p. 1p.</p>	<p>(*) Se acordă câte 2p. pentru fiecare aspect specific (algoritm de bază pentru determinare a valorii maxime dintr-o serie de valori, determinare a poziției unui maxim într-o serie de valori, valori suport verificate pentru determinarea unui maxim pentru fiecare secțiune) conform cerinței.</p>
3.	<p><b>a. Pentru răspuns corect</b> - descriere coerentă a algoritmului, conform cerinței (*) - elemente de eficiență justificate, conform cerinței</p> <p><b>b. Pentru program corect</b> - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - valoare determinată, conform cerinței (*),(**) - eficiență a algoritmului, conform cerinței (***) - variabile declarate, afișare a datelor conform cerinței, corectitudine globală a programului<sup>1)</sup></p>	<p><b>2p.</b> 1p. 1p. 8p. 1p. 5p. 1p. 1p.</p>	<p>(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principal corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar. O soluție posibilă utilizează doi vectori de frecvență, pf și sf, în care pf[i] memorează numărul de apariții ale prefixului i în numerele din fișier, iar sf[i] memorează numărul de apariții ale sufixului i în numerele din fișier. Pe parcursul citirii datelor, se determină, pentru fiecare număr x citit (<math>x &gt; 99</math>), sufixul (<math>x \% 100</math>), respectiv prefixul (<math>[x/10]</math> sau <math>[x/100]</math>, în funcție de numărul de cifre ale lui x) și se actualizează corespunzător cei doi vectori de frecvență. După citirea tuturor numerelor se contorizează toate valorile <math>x \in [10, 99]</math> pentru care <math>sf[x] = pf[x]</math> și <math>sf[x] \neq 0</math>.</p>

<sup>1)</sup> Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.