



**SUBIECTUL al II-lea**

**(30 de puncte)**

**Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Un arbore cu 9 noduri, numerotate de la 1 la 9, este reprezentat prin vectorul de „tați” (3, 3, 0, 5, 2, 5, 2, 5, 8). Descendenții direcți („fii”) ai nodului cu eticheta 5 sunt: **(4p.)**
  - a. 2 7
  - b. 2 8
  - c. 3 3 0
  - d. 4 6 8
2. Numărul de noduri ale unui graf neorientat fără cicluri, cu 26 de muchii și 12 componente conexe este: **(4p.)**
  - a. 18
  - b. 28
  - c. 38
  - d. 48

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

3. Variabila `d`, declarată alăturat, memorează în câmpul `mic` cel mai mic divizor, strict mai mare decât 1, al numărului natural din intervalul  $[2, 10^2]$ , memorat în câmpul `nr`.  

```
struct divizor  
{ int nr, mic;  
  } d;
```

Scrieți o secvență de instrucțiuni în urma executării căreia, pentru numărul memorat în câmpul `nr` al variabilei `d`, se afișează pe ecran mesajul `prim`, dacă numărul este prim, mesajul `patrat` dacă numărul este pătratul unui număr prim, sau două numere naturale, separate printr-un spațiu, reprezentând cel mai mic și cel mai mare dintre divizorii proprii pozitivi ai săi. Divizorii proprii pozitivi ai unui număr sunt divizori pozitivi diferiți de 1 și de el însuși.

**Exemplu:** dacă în câmpul `nr` se memorează numărul 12, iar în câmpul `mic` se memorează numărul 2, se afișează pe ecran

2 6

iar dacă în câmpul `nr` se memorează numărul 9, iar în câmpul `mic` se memorează numărul 3, se afișează pe ecran mesajul

`patrat` **(6p.)**
4. Variabilele `i` și `j` sunt de tip întreg, iar variabila `a` memorează un tablou bidimensional cu 9 linii și 9 coloane, numerotate de la 1 la 9, având inițial toate elementele nule.  
Fără a utiliza alte variabile, scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila `a` să memoreze tabloul alăturat.  

```
for(i=1;i<=9;i++)  
  for(j=1;j<=9;j++)  
    .....
```

**(6p.)**

1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2
1	2	3	3	3	3	3	3	3
1	2	3	4	4	4	4	4	4
1	2	3	4	5	5	5	5	5
1	2	3	4	5	6	6	6	6
1	2	3	4	5	6	7	7	7
1	2	3	4	5	6	7	8	8
1	2	3	4	5	6	7	8	9
5. Un text are cel mult 100 de caractere și este format din cuvinte, numere naturale și spații. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Cuvintele și numerele sunt separate prin câte un spațiu, ca în exemplu.  
Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și afișează pe ecran numărul din text care începe cu cea mai mare cifră, ca în exemplu. Dacă există mai multe astfel de numere, se afișează doar unul dintre acestea, iar dacă textul nu conține niciun număr, se afișează pe ecran mesajul `nu exista`.  
**Exemplu:** pentru textul  
`am 190 de nuci si 70 de castane`  
se afișează  
70 **(10p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Subprogramele `f1` și `f2` sunt definite mai jos.

```
int f1 (int x, int y)
{ if(x%2!=0 || y%2!=0) return 1;
  else return 2*f1(x/2,y/2);
}
```

```
int f2 (int x, int y)
{ if (x==y) return x;
  else
    if(x>y) return f2(x-y,y);
    else return f2(x, y-x);
}
```

Cel mai mare divizor comun al lui 30 și 50 se obține în urma apelului:

**(4p.)**

- a. `f1(30,50)`      b. `f2(30,50)`      c. `f1(30/2,50)`      d. `f2(30/2,50)`

**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Utilizând metoda backtracking, se generează toate posibilitățile de a forma cutii cu bomboane de tipuri distincte din mulțimea {fondante, caramele, dropsuri, acadele}. Într-o cutie sunt cel puțin două tipuri de bomboane, dar nu pot fi și dropsuri și acadele simultan. Două cutii sunt distincte dacă ele conțin cel puțin un tip diferit de bomboane. Primele patru soluții generate sunt, în această ordine, (fondante, caramele), (fondante, caramele, dropsuri), (fondante, caramele, acadele), (fondante, dropsuri). Scrieți a cincea și a șasea soluție, în ordinea generării acestora. **(6p.)**
3. Un număr natural este numit **echilibrat** dacă suma cifrelor sale de pe poziții pare este un număr par, iar suma cifrelor sale de pe poziții impare este un număr impar. Pozițiile cifrelor sunt numerotate de la dreapta la stânga, astfel: cifra unităților este pe poziția 0, cifra zecilor este pe poziția 1 ș.a.m.d. Subprogramul `echilibrat` are un singur parametru, `n`, prin care primește un număr natural ( $n \in [10, 10^9]$ ). Subprogramul returnează valoarea 1 dacă `n` este echilibrat sau valoarea 0 în caz contrar. Scrieți definiția completă a subprogramului. **Exemplu:** dacă `n=25163912`, subprogramul returnează valoarea 1, iar dacă `n=11211`, subprogramul returnează valoarea 0. **(10p.)**
4. Numim **secvență încadrată** a unui șir de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, subșir care începe și se termină cu aceeași valoare. Lungimea secvenței este egală cu numărul de termeni ai acesteia. Fișierul `bac.txt` conține un șir de cel puțin două și cel mult  $10^6$  numere naturale din intervalul  $[0, 9]$ . Numerele sunt separate prin câte un spațiu. În șir există cel puțin doi termeni egali. Se cere să se determine secvențele încadrate din acest șir care au lungime maximă și să se afișeze pe prima linie a ecranului lungimea maximă determinată, iar pe următoarea linie, pentru fiecare astfel de secvență, valoarea primului său termen. Numerele de pe a doua linie sunt afișate în ordine strict crescătoare, separate prin câte un spațiu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare. **Exemplu:** dacă fișierul `bac.txt` conține numerele
- 3 1 5 2 4 5 5 2 5 9 5 7 4 6 8 0 8
- atunci pe ecran se afișează valorile:
- 9
- 4 5
- a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**
- b) Scrieți programul C/C++ corespunzător algoritmului descris. **(8p.)**

**Examenul de bacalaureat național 2018**  
**Proba E. d)**  
**Informatică**

**Barem de evaluare și de notare**  
**(comun pentru limbajele C/C++ și Pascal)**

**Varianta 9**

**Filiera teoretică, profilul real, specializările: matematică-informatică**  
**matematică-informatică intensiv informatică**  
**Filiera vocațională, profilul militar, specializarea matematică-informatică**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct.
- Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.
- Se vor lua în considerare atât implementările concepute pentru compilatoare pe 16 biți, cât și cele pentru compilatoare pe 32 de biți.

**SUBIECTUL I**

**(30 de puncte)**

1.	d	4p.	
2.	a) Răspuns corect: 2	6p.	
	b) Pentru răspuns corect	4p.	Se acordă câte 1p. pentru fiecare dintre cele patru numere din intervalul cerut, valori de forma $4 \cdot k$ , unde $k$ este număr impar.
	c) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului <sup>1)</sup>	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă de tipul indicat, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
	d) Pentru program corect -declarare a tuturor variabilelor -citire conform cerinței -afișare conform cerinței -instrucțiune de decizie conform cerinței -instrucțiuni repetitive conform cerinței (*) -atribuiri conform cerinței -corectitudine globală a programului <sup>1)</sup>	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiuni este corectă.

**SUBIECTUL al II - lea**

**(30 de puncte)**

1.	d	4p.	
2.	c	4p.	
3.	Pentru rezolvare corectă -acces la câmpul unei înregistrări -determinare și afișare a divizorului cerut -tratare a cazului <b>prim</b> -tratare a cazului <b>patrat</b> -corectitudine globală a secvenței <sup>1)</sup>	6p. 1p. 2p. 1p. 1p. 1p.	

Probă scrisă la informatică

Varianta 9

Filiera teoretică, profilul real, specializările: matematică-informatică, matematică-informatică intensiv informatică

Filiera vocațională, profilul militar, specializarea matematică-informatică

Barem de evaluare și de notare

4.	<b>Pentru rezolvare corectă</b>	6p.	Se acordă câte 2p. pentru atribuirea valorilor indicate elementelor din fiecare zonă specifică a tabloului (diagonala principală, sub diagonala principală, deasupra diagonalei principale).
5.	<b>Pentru program corect</b> -declaraire a unei variabile care să memoreze un șir de caractere -citire a șirului -determinare a numărului cerut (*) -afișare date și tratare a cazului <b>nu exista</b> -declaraire variabile simple, corectitudine globală a programului <sup>1)</sup>	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare cuvânt/număr din șir, comparare numere în funcție de prima cifră/lexicografic, algoritm principal corect de determinare a unei valori maxime).

**SUBIECTUL al III - lea**

**(30 de puncte)**

1.	<b>b</b>	4p.	
2.	<b>Răspuns corect:</b> (fondante, acadele) (caramelle, dropsuri)	6p.	Se acordă câte 2p. pentru fiecare aspect al cerinței (prima soluție scrisă, a doua soluție scrisă, ordinea soluțiilor)
3.	<b>Pentru subprogram corect</b> -structură antet principal corectă pentru un subprogram de tipul cerut -declaraire a parametrului -verificare a proprietății cerute (*) -returnare a rezultatului -declaraire a tuturor variabilelor locale, corectitudine globală a subprogramului <sup>1)</sup>	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței (identificare a unei cifre a unui număr, identificare a poziției unei cifre, algoritm principal corect pentru determinarea unei sume, cifre suport pentru cele două sume, paritate poziții, paritate sume).
4.	<b>a)</b> <b>Pentru răspuns corect</b> -coerență a explicării metodei (*) -justificare a unor elemente de eficiență, conform cerinței	2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă metoda aleasă nu este eficientă.
	<b>b)</b> <b>Pentru program corect</b> -operații cu fișiere: declaraire, pregătire în vederea citirii, citire din fișier -determinare a valorilor cerute (*, **) -utilizare a unui algoritm eficient (***) -afișare a datelor conform cerinței, declaraire a tuturor variabilelor, corectitudine globală a programului <sup>1)</sup>	8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă soluția propusă nu prezintă elemente de eficiență. (**) 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 apariții, p și u, în care pentru fiecare valoare i citită din fișier se actualizează p <sub>i</sub> , prima poziție pe care este întâlnită valoarea i, respectiv u <sub>i</sub> , ultima poziție pe care este întâlnită valoarea i. Numerele cerute corespund valorilor i pentru care u <sub>i</sub> -p <sub>i</sub> +1 are valoare maximă.

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