

Examenul de bacalaureat național 2019  
Proba E. d)  
Informatică  
Limbajul C/C++

Varianta 1

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.

1. Variabilele  $x$  și  $y$  sunt întregi. Indicați expresia C/C++ echivalentă cu cea alăturată.  $(x < 3) \ \&\& \ (y \geq 5)$

a.  $!((x < 3) \ || \ (y \geq 5))$

b.  $!(x \geq 3) \ \&\& \ (y < 5)$

c.  $!(!(x < 3) \ || \ !(y \geq 5))$

d.  $!(x \geq 3) \ \&\& \ (y < 5)$

2. Subprogramul  $f$  este incomplet definit alăturat. Indicați expresia cu care pot fi înlocuite punctele de suspensie, astfel încât, în urma apelului de mai jos, să se afișeze cel mai mare divizor comun al numerelor nenule memorate în variabilele întregi  $x$  și  $y$ .  
 $f(x, y, x);$

```
void f(int m, int n, int d)
{
    if(n%d==0 && m%d==0)
        cout<<d; | printf("%d", d);
    else f(.....);
}
```

a.  $m, n, d+1$

b.  $m, n, d-1$

c.  $m+1, n+1, d$

d.  $m-1, n-1, d$

3. Utilizând metoda backtracking, se generează toate băuturile obținute amestecând sucurile a cel puțin două fructe distincte din mulțimea {afine, caise, lămâi, mere, pere}. Primele cinci soluții obținute sunt, în această ordine: (afine, caise), (afine, caise, lămâi), (afine, caise, lămâi, mere), (afine, caise, lămâi, mere, pere) și (afine, caise, lămâi, pere). A șasea soluție este:

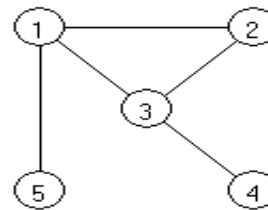
a. (afine, caise, mere)

b. (afine, caise, mere, pere)

c. (afine, mere, pere)

d. (afine, lămâi, mere, pere)

4. Indicați un lanț elementar în graful neorientat cu 5 noduri, numerotate de la 1 la 5, reprezentat alăturat.



a. 1, 2, 3, 1

b. 1, 2, 3, 4

c. 1, 2, 3, 4, 5

d. 1, 3, 2, 1, 5

5. Indicați valorile ce pot reprezenta numărul de fii ai fiecăruia dintre cele șase noduri ale unui arbore cu rădăcină.

a. 0, 0, 0, 1, 2, 3

b. 0, 0, 0, 0, 2, 3

c. 0, 0, 0, 3, 3, 3

d. 1, 1, 1, 1, 1, 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  $c$ .

- a) Scrieți valoarea afișată dacă se citește, în această ordine, numerele 10 și 3. **(6p.)**
- b) Dacă pentru  $k$  se citește numărul 5, scrieți trei numere care pot fi citite pentru  $n$  astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, valoarea afișată să fie 10. **(6p.)**
- c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

```

citește n, k
(numere naturale, k număr prim)
p ← 0; i ← 1
cât timp i ≤ n execută
    x ← i
    cât timp x % k = 0 execută
        x ← [x/k]; p ← p + 1
    i ← i + 1
scrie p
    
```

- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind prima structură `cât timp...execută` cu o structură de tip `pentru...execută`. **(6p.)**

- 2. Variabila  $e$ , declarată alăturat, memorează informații despre un eveniment din anul 2019 (numărul de ordine și data desfășurării sale), iar variabila  $d$  memorează o dată calendaristică din același an. Scrieți o expresie care are valoarea 1 dacă și numai dacă data memorată în variabila  $d$  este anterioară datei desfășurării evenimentului corespunzător variabilei  $e$ . **(6p.)**

```

struct tdata
{ int zi, luna;
}d;
struct eveniment
{ int nr;
  struct tdata dev;
}e;
    
```

- 3. Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, având inițial toate elementele nule.

Fără a utiliza alte variabile decât cele menționate, 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<=5; i++)
    for (j=1; j<=5; j++)
        .....
    
```

```

4 4 4 4 4
4 3 3 3 3
4 3 2 2 2
4 3 2 1 1
4 3 2 1 0
    
```

**(6p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

- 1. Subprogramul `Ega1` are un parametru,  $n$ , prin care primește un număr natural cu cel puțin o cifră impară ( $n \in [10, 10^9]$ ). Subprogramul returnează valoarea 1 dacă toate cifrele impare ale lui  $n$  sunt egale între ele sau valoarea 0 în caz contrar. Scrieți definiția completă a subprogramului.

**Exemplu:** dacă  $n=7727470$  sau  $n=7240$  atunci subprogramul returnează 1, iar dacă  $n=7921470$  atunci subprogramul returnează 0. **(10p.)**

- 2. Într-un text cu cel mult 100 de caractere cuvintele sunt formate numai din litere mari și mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul precizat, apoi un număr natural,  $n$  ( $n \in [1, 10^2]$ ), și afișează pe ecran, pe linii separate, cuvintele din text care au exact  $n$  litere. Cuvintele sunt afișate în ordinea apariției lor în text, iar dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul `nu exista`.

**Exemplu:** dacă se citește textul `Fat Frumos este cel mai viteaz`

iar  $n=6$ , se afișează pe ecran:

```

Frumos
viteaz
    
```

**(10p.)**

- 3. Șirul  $f$  este definit astfel:  $f_1=x$ ;  $f_2=y$ ;  $f_3=z$ ;  $f_i=f_{i-1}+f_{i-2}-f_{i-3}$ , unde  $x$ ,  $y$ ,  $z$  și  $i$  sunt numere naturale nenule,  $i > 3$ .

De exemplu, dacă  $x=1$ ,  $y=2$  și  $z=4$  șirul este: 1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, ...

Se citește de la tastatură un număr natural,  $n$  ( $n \in [1, 10^4]$ ), apoi trei numere naturale din intervalul  $[1, 10^2]$ ,  $x$ ,  $y$  și  $z$ , reprezentând, în această ordine, primii trei termeni ai șirului precizat mai sus. Se cere să se scrie în fișierul `bac.txt` primii  $n$  termeni ai șirului, separați prin câte un spațiu, în ordine inversă a apariției lor în șir. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

**Exemplu:** dacă  $n=10$ ,  $x=1$ ,  $y=2$  și  $z=4$  fișierul conține numerele: 14 13 11 10 8 7 5 4 2 1

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

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

Examenul de bacalaureat național 2019

Proba E. d)

Informatică

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

Varianta 1

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

**SUBIECTUL I**

(20 de puncte)

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

**SUBIECTUL al II - lea**

(40 de puncte)

1.	<b>a) Răspuns corect: 4</b>	<b>6p.</b>	
	<b>b) Pentru răspuns corect</b>	<b>6p.</b>	Se acordă câte 2p. pentru fiecare dintre cele trei numere conform cerinței (oricare dintre numerele 45,46,47,48,49).
	<b>c) Pentru program corect</b> -declarare variabile -citire date -afișare date -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului <sup>1)</sup>	<b>10p.</b> 1p. 1p. 1p. 4p. 2p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	<b>d) Pentru algoritm pseudocod corect</b> -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului <sup>1)</sup>	<b>6p.</b> 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principal corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	<b>Pentru rezolvare corectă</b> -acces la câmpurile de pe primul nivel al înregistrării -acces la câmpurile de pe al doilea nivel al înregistrării -verificare a condiției impuse (*) -corectitudine globală a secvenței <sup>1)</sup>	<b>6p.</b> 1p. 1p. 3p. 1p.	(*) Se acordă numai 2p. dacă doar unul dintre cazuri (date în aceeași lună, date în luni diferite) s-a tratat conform cerinței.
3.	<b>Pentru rezolvare corectă</b> -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*) -corectitudine globală a secvenței <sup>1)</sup>	<b>6p.</b> 1p. 4p. 1p.	(*) Se acordă numai 2p. dacă s-au atribuit valori conform cerinței doar elementelor situate sub diagonala principală sau doar elementelor situate deasupra diagonalei principale.

**SUBIECTUL al III - lea**

(30 de puncte)

1.	<b>Pentru subprogram corect</b> -antet subprogram (*) -verificare a proprietății cerute (**) -instrucțiune/instrucțiuni de returnare a rezultatului -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului <sup>1)</sup>	<b>10p.</b> 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametru de intrare) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unei cifre pare/impare, cifre suport, algoritm principal corect de verificare a unei proprietăți).
----	--	---	---

2.	<b>Pentru program corect</b> -declarare a unei variabile care să memoreze un șir de caractere -citire a datelor -determinare a cuvintelor cerute (*) -afișarea a datelor în formatul cerut și tratare a cazului <b>nu exista</b> -declarare a variabilelor simple, corectitudine globală a programului <sup>1)</sup>	<b>10p.</b> 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (obținere a unui cuvânt, determinare a lungimii unui cuvânt, cuvinte suport conform cerinței).
3.	<b>a) Pentru răspuns corect</b> -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență <b>b) Pentru program corect</b> -operații cu fișiere: declarare, pregătire în vederea scrierii, scriere în fișier -determinare a valorilor cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului <sup>1)</sup>	<b>2p.</b> 1p. 1p. <b>8p.</b> 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă generează termenii șirului deducând forma generală a unui astfel de termen (pentru $i \geq 1$ ): dacă $i$ este par, $f_i = (i/2 - 1) \cdot (z - x) + y$ , iar dacă $i$ este impar, $f_i = [i/2] \cdot z - ([i/2] - 1) \cdot x$ .

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