

Examenul de bacalaureat național 2020  
Proba E. d)  
INFORMATICĂ  
Limbaajul C/C++

MODEL

*Filieră teoretică, profil real, specializare științe ale naturii*

- 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**, **y** și **z** sunt de tip întreg și memorează numere naturale din intervalul  $[1, 10^3]$ . Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă valoarea variabilei **x** este strict mai mică decât valoarea oricăreia dintre variabilele **y** și **z**.

- a. **`z+x<x+y && x+z>z+y`** b. `z+x<x+y && z+y>y+x`  
c. **`x+z<z+y && z+y>y+x`** d. `x+y<y+z && x+z>z+y`

2. În secvențele notate cu **s1**, **s2** și **s3**, definite mai jos, toate variabilele sunt întregi.

```
//s1
p=1;
for (i=2; i<=n; i++)
    p=p*i;
```

```
//s2
p=1;
while (n>1)
{ p=p*n*(n-1);
  n=n-2;
}
```

```
//s3
p=1;
if (n%2==1)
{ p=n; n=n-1; }
for (i=1; i<=n/2; i++)
    p=p*i*(n+1-i);
```

Dacă valoarea inițială a variabilei **n** este 12, se obține aceeași valoare pentru variabila **p** în urma executării, independent, a secvențelor:

- a. **s1 și s2** b. s1 și s3 c. s2 și s3 d. s1, s2 și s3

3. Variabilele **x** și **y** sunt de tip real. O transcriere în limbajul C/C++ a expresiei  $\sqrt{x \cdot (x^2 - y)}$  alăturată este:

- a. **`pow(x*(sqrt(x)-y))`** b. `sqrt(x*(pow(x,2)-y))`  
c. **`pow(x*x*x-y)`** d. `sqrt(x*x*x-y)`

4. Elementele unui tablou unidimensional sunt, în această ordine, **(2, 5, 6, 12, 55, 57, 79)**. Pentru a verifica dacă în tablou există elementul cu valoarea **x=56**, se aplică metoda căutării binare. Succesiunea de elemente cu care se compară valoarea **x** pe parcursul aplicării metodei indicate este:

- a. **12, 57, 55** b. 12, 55, 57 c. 2, 6, 12, 57 d. 79, 57, 55

5. În urma interclasării în ordine descrescătoare a tablourilor unidimensionale **A** și **B** se obține tabloul **(45, 29, 17, 16, 12, 12, 10, 7, 3, 2)**. Elementele tablourilor, în ordinea în care apar în acesta, pot fi:

- a. **A=(4, 2, 1, 1, 1, 1, 1)**  
    **B=(5, 9, 7, 6, 2, 2, 0, 7, 3, 2)** b. A=(45, 29, 17, 16, 12)  
    **B=(10, 7, 3, 2)** c. A=(22, 14, 8, 8, 6, 6, 5, 3, 1, 1)  
    **B=(23, 15, 9, 8, 6, 6, 5, 4, 2, 1)** d. A=(45, 29, 17, 12, 10, 3)  
    **B=(16, 12, 7, 2)**

**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  $b$ .
- a. Scrieți valoarea afișată în urma executării algoritmului dacă se citește, în această ordine, numerele **21, 38 și 4**. (6p.)
- b. Dacă pentru  $m$  și  $x$  se citesc numerele **20**, respectiv **2020**, scrieți cea mai mică și cea mai mare valoare care pot fi citite pentru variabila  $n$ , astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze **2020**. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structura **repetă . . . până când** cu o structură de alt tip. (6p.)
- ```
citește m, n, x
(numere naturale nenule, m ≤ n)
s ← 0; pm ← 1; pn ← 1
repetă
    dacă m%x=0 atunci
        s ← s+m; pm ← x
    ■
    dacă n%x=0 și m≠n atunci
        s ← s+n; pn ← x
    ■
    m ← m+pm; n ← n-pn
până când m>n
scrie s
```
2. Variabila **id** memorează identificatorul unei zone de parcare dintr-un oraș (o literă a alfabetului englez), iar variabilele întregi **numar** și **pret** memorează numărul de locuri închiriate în această zonă pe parcursul lunii curente, respectiv prețul practicat pentru închirierea unui loc pentru o lună. Declarați corespunzător variabila **id** și scrieți o instrucțiune C/C++ care să afișeze pe ecran suma obținută în urma închirierii pe parcursul lunii curente a tuturor locurilor de parcare din zona precizată. (6p.)
3. În secvența alăturată toate variabilele sunt de tip întreg. Scrieți secvența înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, valoarea variabilei **p** să fie numărul de ordine al ultimului număr **2020** citit, sau **0**, dacă nu există astfel de numere.
- ```
p=.....;
for(i=1; i<=10; i++)
{ cin>>x; | scanf("%d", &x);
.....
}
```
- Exemplu:** dacă se citește, în această ordine, numerele **20, 19, 2020, 15, 2020, 6, 7, 21, 24, 36**, în urma executării secvenței, **p** are valoarea **5**. (6p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Se citește un număr natural,  $n$  ( $n \geq 1$ ), și se cere să se scrie numărul obținut prin duplicarea fiecărei cifre impare a lui  $n$  sau  $-1$  dacă acesta nu are nicio cifră impară. Scrieți, în pseudocod, algoritmul de rezolvare pentru problema enunțată. **Exemplu:** dacă  $n=2019$ , se scrie **201199**. (10p.)
2. Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul  $[0, 10^9]$ , în această ordine: numerele  $n$  și  $x$ , apoi cele  $n$  elemente ale unui tablou unidimensional. Programul modifică ordinea valorilor din tablou, astfel încât primele poziții să fie ocupate de mulțimea formată de cele care sunt mai mari sau egale cu  $x$ , iar următoarele poziții, în continuarea acestora, să fie ocupate de mulțimea celorlalte numere. Valorile din aceeași mulțime sunt memorate într-o ordine oarecare. Elementele tabloului obținut sunt afișate pe ecran, separate prin câte un spațiu, iar dacă una dintre cele două mulțimi este vidă, se afișează pe ecran doar mesajul **nu exista**. **Exemplu:** pentru  $n=9$ ,  $x=19$  și tabloul **(20, 19, 20, 5, 20, 18, 7, 75, 3)**, unul dintre tablourile obținute poate fi: **(20, 19, 20, 20, 75, 5, 18, 7, 3)**. (10p.)
3. Fișierul **numere.in** conține un șir de numere naturale din intervalul  $[1, 10^9]$ . Numerele din șir sunt ordonate strict descrescător și sunt separate prin câte un spațiu. Se cere să se determine numărul valorilor naturale distincte din intervalul închis delimitat de prima și ultima valoare din șir care **NU** se găsesc în șirul menționat mai sus. Numărul determinat se afișează pe ecran. Proiectați un algoritm eficient din punctul de vedere al spațiului de memorie și al timpului de executare. **Exemplu:** dacă fișierul conține numerele  
**8 5 3 2**  
se afișează pe ecran **3** (în șir nu se găsesc valorile **7 6 4**).  
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)

Examenul de bacalaureat național 2020

Proba E. d)  
INFORMATICĂ

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

MODEL

Filieră teoretică, profil real, specializare științe ale naturii

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

SUBIECTUL al II - lea

(40 de puncte)

1.	a) Răspuns corect: 120	6p.	
	b) Răspuns corect: 2020 4039	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței.
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiune repetitivă - instrucțiuni de decizie (*) - atribuirii - corectitudine globală a programului <sup>1)</sup>	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile de decizie este conform cerinței.
	d) 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 conform cerinței, principal corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă - declarare a variabilei conform cerinței - afișare a datelor conform cerinței - corectitudine globală a secvenței <sup>1)</sup>	6p. 2p. 3p. 1p.	
3.	Pentru rezolvare corectă - înlocuire a punctelor de suspensie din prima zonă - înlocuire a punctelor de suspensie din cea de a doua zonă - corectitudine globală a secvenței <sup>1)</sup>	6p. 2p. 3p. 1p.	

SUBIECTUL al III - lea

(30 de puncte)

1.	Pentru algoritm corect - citire a datelor - determinare a numărului cerut (*) - scriere a datelor - scriere principal corectă a structurilor de control (**)	10p. 1p. 6p. 1p. 2p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței (acces la o cifră a unui număr, cifre suport analizate, cifre pare/impare, duplicare a unei cifre, cifre suport duplicate, tratare caz -1). (**) Se acordă punctajul pentru orice formă corectă de structură repetitivă sau decizională.
----	--	----------------------------------	--

2.	<p><b>Pentru program corect</b></p> <ul style="list-style-type: none"> <li>- declarare a unei variabile care să memoreze un tablou</li> <li>- citire a datelor</li> <li>- memorare a numerelor conform cerinței (*)</li> <li>- afișare a datelor și tratare a cazului <b>nu exista</b></li> <li>- declarare a variabilelor simple, corectitudine globală a programului<sup>1)</sup></li> </ul>	<p><b>10p.</b></p> <p>1p.</p> <p>1p.</p> <p>6p.</p> <p>1p.</p> <p>1p.</p>	<p>(*) Se acordă câte 2p. pentru fiecare aspect (memorare pe poziții consecutive a valorilor din aceeași mulțime, ordine a mulțimilor de valori, elemente suport) conform cerinței.</p>
3.	<p><b>a) Pentru răspuns corect</b></p> <ul style="list-style-type: none"> <li>- coerență a descrierii algoritmului (*)</li> <li>- justificare a elementelor de eficiență</li> </ul>	<p><b>2p.</b></p> <p>1p.</p> <p>1p.</p>	<p>(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient.</p>
	<p><b>b) Pentru program corect</b></p> <ul style="list-style-type: none"> <li>- operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier</li> <li>- determinare a valorii cerute (*),(**)</li> <li>- utilizare a unui algoritm eficient (***)</li> <li>- declarare a variabilelor, afișare a datelor, corectitudine globală a programului<sup>1)</sup></li> </ul>	<p><b>8p.</b></p> <p>1p.</p> <p>5p.</p> <p>1p.</p> <p>1p.</p>	<p>(**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare.</p> <p>(***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria.</p> <p>O soluție posibilă utilizează un contor, inițializat cu 0, apoi parcurge numerele din fișier memorând, la fiecare pas, ultimele două numere citite, x și y, și actualizează contorul adăugând valoarea x-y-1.</p>

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