

SIMULARE EXAMENUL DE BACALAUREAT – 2025
Proba E. d) - Proba scrisă la INFORMATICĂ
Limbajul C/C++

Simulare

Filieră teoretică, profil real, specializare științe ale naturii

- 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.

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.

- Precizați ce valoare va avea variabila reală x , după executarea următoarei instrucțiuni **C/C++**.
 $x = 7.51 + 35/4 * 67\%8 - 2.83;$
 - Instrucțiunea este incorectă
 - 4
 - 4.68
 - 28.68
- Se consideră tabloul $x = (23, 16, 10, 25, 17)$. Pentru fiecare element din vector se calculează numărul de divizori și se rearanjează elementele din tablou, astfel încât să fie dispuse în ordine descrescătoare în funcție de numărul de divizori. Care este tabloul obținut după ordonare?
 - (25, 23, 17, 16, 10)
 - (16, 10, 25, 23, 17)
 - (17, 23, 25, 10, 16)
 - (10, 16, 17, 23, 25)
- Aplicând algoritmul de căutare binară pentru căutarea unei valori x în tabloul unidimensional (19,15,11,9,8,7,1) se fac exact 3 comparații. Care dintre următoarele valori poate fi valoarea variabilei x ?
 - 19, 11
 - 15, 7
 - 8, 1
 - 19, 11, 8, 1
- Indicați expresia **C/C++** care are valoarea 1.
 - `ceil(30.19) == ceil(30.91)`
 - `ceil(30.19) == floor(30)`
 - `ceil(30.91) == floor(30.19)`
 - `ceil(30.19) == floor(30.91)`
- Se consideră un tablou x format din 10 elemente, numere naturale, și variabilele i și s de tip întreg. Completați punctele de suspensie din secvența de program alăturată astfel încât să se afișeze suma elementelor tabloului x , divizibile cu 2 și cu 7.


```

s=0;
for (i=0; i<10; i++)
    if( ..... )
        s=s+x[i];
cout<<s;
      
```

 - `i%2==0 && i%7==0`
 - `x[i]%2==0 || x[i]%7==0`
 - `x[i]%2==0 && x[i]%7==0`
 - `x[i]%2!=0 && x[i]%7!=0`

SUBIECTUL II**(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 întreagă a numărului real c .

- Scrieți numărul afișat în urma executării algoritmului dacă pentru a se citește valoarea 5 iar pentru n valoarea 4. (6p.)
- Scrieți două seturi de date distincte care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze valoarea 64. (6p.)
- Scrieți programul **C/C++** corespunzător algoritmului dat. (10p.)
- Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura repetitivă **cât timp...** **execută** cu o structură repetitivă de alt tip. (6p.)

```

citește a, n (numere naturale
nenule)
p ← 1
cât timp n ≠ 0 execută
    c ← n % 2
    dacă c=1 atunci
        p ← p * a
    n ← [n / 2]
    a ← a * a
scrie p
      
```

2. În secvența de mai jos variabilele **i** și **j** sunt de tip întreg. 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, să se afișeze numerele de mai jos:

0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
3 4 5 6 7
4 5 6 7 8

for (i = 0; i < 5; i++)
for(j = 0; j < 5; j++)

.....

(6p.)

3. Variabilele reale distincte **h1**, **h2**, **h3** memorează înălțimea a trei elevi, iar variabilele întregi **nrm1**, **nrm2**, **nrm3** memorează numerele matricole ale celor trei elevi. Declarați variabilele și scrieți o secvență de instrucțiuni în urma căreia să se afișeze pe ecran numărul matricol al elevului care are înălțimea cea mai mare.

Exemplu: dacă **h1=1.57**, **h2=1.70**, **h3=1.58**,

nrm1=101, **nrm2=105**, **nrm3=109** se va afișa valoarea **105**.

(6p.)

SUBIECTUL III

(30 de puncte)

1. În vacanța de iarnă Alex a primit un joc captivant și astfel a tot amânat lecturile de vacanță, fapt pentru care în prima săptămână din modulul al III-lea a primit o notă mică la școală. Supărat, tatăl său i-a parolat jocul cu un cod de acces. Pentru că a început să recupereze din lecturi, tatăl îi dă un indiciu pentru obținerea codului de acces: îi arată un număr și îi spune că numărul format din prima și ultima cifră ale numărului dat reprezintă codul de acces.

Scrieți un algoritm în pseudocod care citește un număr natural **n** de minimum 2 și maximum 9 cifre reprezentând numărul primit de Alex de la tatăl său și construiește numărul natural de două cifre și apoi afișează codul de acces.

Exemplu: dacă **n=274025**, atunci, se va afișa 25.

(10p.)

2. Scrieți un program C/C++ care citește de la tastatură un număr **n** ($n \in [2, 10^2]$) și cele **2*n** elemente ale unui tablou unidimensional, numere naturale din intervalul **[1, 10⁶]**. Programul transformă tabloul în memorie, interschimbând prima valoare pară din șirul primelor **n** elemente cu prima valoare impară din șirul ultimelor **n** elemente și afișează pe ecran separate printr-un spațiu elementele tabloului. În cazul în care unul din elementele care urmează a fi interschimbate nu există, tabloul rămâne nemodificat.

Exemplu: **n=5**

7 8 9 19 6 8 9 11 4 3 Se obține tabloul 7 9 9 19 6 8 8 11 4 3

(10p.)

3. Se consideră o banchiză pe care sunt așezați, în linie, mai mulți pinguini. Se cunosc înălțimile acestora, exprimate prin numere naturale. Se știe că pinguinii **Imperiali** sunt cei a căror înălțime este cuprinsă între două valori date **a** și **b** (adică valori aflate în intervalul **[a, b]**).

Se cere să se determine secvența de lungime maximă formată numai din pinguini **Imperiali** și numărul de secvențe de lungime maximă. O secvență este un șir de cel puțin 2 elemente din linie aflate pe poziții consecutive.

Fișierul **bac.txt** conține maximum 10⁵ numere naturale de cel mult 4 cifre fiecare: pe prima linie valorile **a** și **b** iar pe următoarea linie un șir de numere naturale reprezentând înălțimile pinguinilor de pe banchiză. Se vor afișa pe ecran, separate printr-un spațiu, două valori **lgmax** și **nr** reprezentând lungimea maximă a secvenței și numărul de secvențe de lungime maximă dacă aceasta există, sau valoarea **-1** în caz contrar.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu1: dacă fișierul conține valorile alăturate, atunci pe ecran se afișează 3 3

bac.txt

7 10
1 2 7 8 10 4 5 8 8 9 11 12 9 9 10 12

Exemplu2: dacă fișierul conține valorile alăturate, atunci pe ecran se afișează -1

bac.txt

7 10
1 2 3 4 5 6 7 11 8

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

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



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

Simulare

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 (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)**

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

SUBIECTUL II **(40 de puncte)**

1.	a) Răspuns corect: 625	6p.	
	b) Pentru răspuns corect	6p.	Se acordă câte 3 p. pentru fiecare dintre cele doua perechi de numere conform cerinței (oricare dintre perechile de numere 64 1, 8 2, 4 3, 2 6).
	c) Pentru program corect -declaraire variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive -atribuiri -corectitudine globală a programului ¹	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	
	d) Pentru algoritm pseudocod corect --utilizare a unei structuri repetitive cu test final (*) -expresie logică pentru test final în cadrul secvenței obținute prin înlocuire, conform cerinței -algoritm complet, corectitudine globală a algoritmului ¹	6p. 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 (repeta ...pană când, executăcât timp, do while etc.).
2.	Pentru rezolvare corectă -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*) -corectitudine globală a secvenței ¹	6p. 1p. 4p. 1p.	*) Se acordă câte 2p. pentru fiecare aspect specific (mulțime a valorilor suport utilizate, corespondență valori poziții) conform cerinței.
3.	Pentru rezolvare corectă -declaraire a variabilelor conform cerinței -determinare a ordinii valorilor cerute (*) - corectitudine globală a secvenței ¹	6p. 1p. 4p. 1p.	*) Se acordă numai 2p. dacă sunt conform cerinței doar o parte dintre condițiile necesare determinării valorii cerute.

SUBIECTUL III **(30 de puncte)**

1.	Pentru algoritm corect -citire a datelor -determinare a valorii cerute (*) -afișare a datelor -scriere principal corectă a structurilor de control, corectitudine globală a algoritmului ¹) (**)	10p. 1p. 6p. 1p. 2p.	(*) Se acordă câte 2p. pentru fiecare aspect (identificare a ultimei cifre, identificare a primei cifre, formarea unui număr cu cifrele identificate, conform cerinței. (**) Se va puncta orice formă corectă de structură repetitivă sau decizională.
-----------	--	---	---



2.	Pentru program corect - declarare a unei variabile care să memoreze un tablou unidimensional - citire a datelor - construire a tabloului conform cerinței (*) - afișare a datelor - declarare a variabilelor simple, corectitudine globală a programului¹⁾	10p. 1p. 1p. 6p. 1p. 1p.	*) Se acordă câte 1p. pentru fiecare aspect specific (identificare a unui element par/impar, identificare a poziției primului element par situat în zona indicată, identificare a poziției primului element impar situat în zona indicată, identificarea cazului de interschimbare a celor două elemente, interschimbarea elementelor)
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 scrierii - determinare a valorilor cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, citire a datelor, corectitudine globală a programului¹⁾	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) 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 linear care utilizează eficient memoria. O soluție posibilă citește din fișier primele două valori și apoi pe rând celelalte valori verificând la fiecare pas dacă valoarea curentă poate face parte din secvență ($x \in [a, b]$) actualizând lungimea acesteia, dacă nu îndeplinește condiția se actualizează lungimea maximă și numărul de apariții a lungimii maxime.

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