

**Examenul de bacalaureat național 2020**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

**Testul 3**

*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.

- Variabilele  $x$  și  $y$  sunt de tip întreg și memorează numere naturale nenule. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în  $x$  are cifra unităților egală cu 2 și este un multiplu al numărului memorat în  $y$ .
  - $x \% 2 == 10 \ || \ y \% x == 0$
  - $x \% 10 == 2 \ || \ y \% x == 0$
  - $x \% 2 == 10 \ \&\& \ x \% y == 0$
  - $x \% 10 == 2 \ \&\& \ x \% y == 0$
- Subprogramul `f` este definit alăturat. Indicați ce se afișează în urma apelului de mai jos.  
`f(3);`

```
void f(int x)
{ cout<<x; | printf("%d", x);
  while(x>0) { f(x-1); x=x-1; }
}
```

  - 321021010
  - 32100100
  - 3210
  - 321
- Utilizând metoda backtracking se generează toate posibilitățile de a forma șiraguri din câte 4 pietre prețioase din mulțimea {`rubin, opal, safir, smarald, topaz`}, astfel încât pe oricare două poziții alăturate să nu se afle două pietre din submulțimea {`rubin, safir, topaz`}. Primele opt șiraguri generate sunt, în această ordine, (`rubin, opal, rubin, opal`), (`rubin, opal, rubin, smarald`), (`rubin, opal, opal, rubin`), (`rubin, opal, opal, opal`), (`rubin, opal, opal, safir`), (`rubin, opal, opal, smarald`), (`rubin, opal, opal, topaz`), (`rubin, opal, safir, opal`). Ultimul șirag generat este:
  - (`topaz, smarald, topaz, topaz`)
  - (`topaz, smarald, topaz, opal`)
  - (`topaz, smarald, topaz, smarald`)
  - (`topaz, smarald, smarald, topaz`)
- Un arbore cu 10 noduri, numerotate de la 1 la 10, este reprezentat prin vectorul de „tați”  $t = (2, 5, 1, 1, 0, 3, 3, 7, 4, 6)$ . Indicați numărul de frunze ale arborelui.
  - 3
  - 4
  - 5
  - 6
- Un graf neorientat are 10 noduri, numerotate de la 1 la 10, și muchiile [1,2], [2,3], [2,10], [3,10], [4,5], [4,6], [5,6], [6,9], [7,8], [7,9], [8,9]. Indicați numărul minim de muchii care trebuie adăugate pentru ca graful obținut să fie eulerian.
  - 1
  - 2
  - 3
  - 4

**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 întreagă a numărului real  $c$ .
- a. Scrieți ce se afișează dacă se citește valoarea 2754578. (6p.)
- b. Scrieți două numere care pot fi citite astfel încât, în urma executării algoritmului pentru primul dintre ele să se afișeze 1 0, iar pentru al doilea să se afișeze -1 0. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structura **repetă...până când** cu o structură repetitivă de alt tip. (6p.)
- ```

citește n (număr natural, n>9)
c1←n%10; n←[n/10]; c2←n%10
dacă c1=c2 atunci s←0
altfel
  dacă c1>c2 atunci s←1
  altfel s←-1
  ■
  ■
repetă
  c1←n%10; n←[n/10]; c2←n%10
până când (c1-c2)*s≤0
scrie s, ' ',n
  
```
2. Variabila **fig** memorează date specifice unui cerc: coordonatele reale (abscisa și ordonata), în planul  $xOy$ , ale centrului cercului, precum și lungimea razei acestuia. Știind că expresiile C/C++ de mai jos au ca valori numere reale reprezentând datele specifice ale cercului, scrieți definiția unei structuri cu eticheta **cerc**, care permite memorarea datelor precizate, și declarați corespunzător variabila **fig**.  
**fig.centru.x      fig.centru.y      fig.raza** (6p.)
3. Variabila **p** este de tip întreg, iar variabila **s** memorează un șir de cel mult 20 de caractere, numai litere mari ale alfabetului englez. Fără a utiliza alte variabile, scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran toate literele șirului memorat de variabila **s**, cu excepția vocalei **A**, dacă în șirul inițial aceasta este alături de vocala **I**. Literele se afișează în ordinea apariției lor în șir.  
**Exemplu:** dacă șirul memorat în variabila **s** este **ALIANTA** sau **ALAINTA** se va afișa **ALINTA**. (6p.)

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Subprogramul **factori** are doi parametri,  $n$  și  $m$ , prin care primește câte un număr natural din intervalul  $[1, 10^9]$ . Subprogramul returnează numărul valorilor prime care se regăsesc atât în descompunerea în factori primi a lui  $n$ , cât și în descompunerea în factori primi a lui  $m$ .  
Scrieți definiția completă a subprogramului.  
**Exemplu:** dacă  $n=750$  și  $m=490$ , atunci subprogramul returnează 2 ( $750=2 \cdot 3 \cdot 5^3$ ,  $490=2 \cdot 5 \cdot 7^2$ ). (10p.)
2. Scrieți un program C/C++ care citește de la tastatură un număr natural,  $n$  ( $n \in [2, 20]$ ), și construiește în memorie un tablou bidimensional cu  $n$  linii și  $n$  coloane, având proprietățile:
- toate elementele situate pe diagonala secundară sunt nule;
  - fiecare linie conține, începând cu diagonala secundară, de la dreapta la stânga, un șir strict crescător de numere consecutive, iar începând cu diagonala secundară, de la stânga la dreapta, tot un șir strict crescător de numere consecutive.
- Programul afișează pe ecran tabloul construit, fiecare linie a tabloului pe câte o linie a ecranului, cu elementele aflate pe aceeași linie separate prin câte un spațiu.
- Exemplu:** dacă  $n=5$  se afișează pe ecran tabloul alăturat. (10p.)
- ```

4 3 2 1 0
3 2 1 0 1
2 1 0 1 2
1 0 1 2 3
0 1 2 3 4
  
```
3. Fișierul **bac.in** conține un șir de cel mult  $10^6$  numere întregi din intervalul  $[-10^9, 10^9]$ , separate prin câte un spațiu. Cel puțin un număr din șir este negativ.  
Se cere să se afișeze pe ecran lungimea maximă a unei secvențe a șirului care fie începe, fie se încheie cu un număr negativ. O secvență este formată din termeni aflați pe poziții consecutive în șir, iar lungimea secvenței este egală cu numărul de termeni ai acesteia. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.  
**Exemplu:** dacă fișierul conține numerele 12 25 -6 7 80 -75 101 -6 52 -124 87 99 210 pe ecran se afișează 11 (corespunzător secvenței -6 7 80 -75 101 -6 52 -124 87 99 210).
- a. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)
- b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

Examenul de bacalaureat național 2020  
Proba E. d)  
Informatică

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

Testul 3

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)

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

SUBIECTUL al II - lea (40 de puncte)

1.	a) Răspuns corect: 1 275	6p.	Se acordă câte 3p. pentru fiecare dintre cele două valori conform cerinței.
	b) Pentru răspuns corect	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței (orice număr cu cifre în ordine strict crescătoare, respectiv orice număr cu cifre în ordine strict descrescătoare).
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni de decizie (*) -instrucțiune repetitivă -atribuiri -corectitudine globală a programului <sup>1)</sup>	10p. 1p. 1p. 1p. 3p. 2p. 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, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -definire a structurii/înregistrării (*) -declarare a variabilei conform cerinței -corectitudine globală a secvenței <sup>1)</sup>	6p. 3p. 2p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (definire principial corectă a unei structuri/înregistrări, câmpuri, etichetă/nume) conform cerinței.
3.	Pentru rezolvare corectă -acces la un caracter al șirului -afișare a valorilor conform cerinței (*) -corectitudine globală a secvenței <sup>1)</sup>	6p. 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect specific (identificare a literelor A și I, identificare a succesiunii de litere A I, identificare a succesiunii de litere I A, litere suport) conform cerinței.

SUBIECTUL al III - lea (30 de puncte)

1.	Pentru subprogram corect -antet subprogram (*) -determinare a valorilor cerute (**) -instrucțiune/instrucțiuni de returnare a rezultatului -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului <sup>1)</sup>	10p. 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri de intrare) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unui divizor, divizori primi comuni, algoritm principial corect de numărare).
----	---	----------------------------------	--

2.	<p><b>Pentru program corect</b> -declarare a unei variabile care să memoreze un tablou bidimensional -citire a datelor -construire a tabloului conform cerinței (*) -afișare a datelor -declarare a variabilelor simple, corectitudine globală a programului<sup>1)</sup></p>	<p><b>10p.</b> 1p. 1p. 6p. 1p. 1p.</p>	<p>(*) Se acordă câte 1p. pentru fiecare aspect specific (identificare a unui element situat pe diagonala secundară, atribuire valori pe diagonala secundară, construire a unui șir crescător, construire a unui șir descrescător, valori suport, construire în memorie) conform cerinței.</p>
3.	<p><b>a) Pentru răspuns corect</b> -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență</p> <p><b>b) Pentru program corect</b> -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorilor cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului<sup>1)</sup></p>	<p><b>2p.</b> 1p. 1p.</p> <p><b>8p.</b> 1p. 5p. 1p. 1p.</p>	<p>(*) 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 parcurge șirul din fișier, numără valorile citite în variabila nr și memorează poziția primului număr negativ, p1, precum și poziția ultimului număr negativ, p2. Valoarea cerută este cea mai mare dintre valorile nr-p1+1 și p2.</p>

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