

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

Testul 6

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.

- Expresia C/C++
 $(x \geq 18) \ \&\& \ ! (x < 19 \ || \ x > 20) \ \&\& \ (x \leq 21)$
are valoarea 1 dacă și numai dacă valoarea memorată de variabila întreagă x aparține intervalului:
a. [18,20] b. [18,21] c. [19,20] d. [19,21]
- Utilizând metoda backtracking se generează toate posibilitățile de a planta de-a lungul unei alei cinci arbuști decorativi din mulțimea {caprifoi, iasomie, liliac, tamarix, scumpie}. Două soluții sunt diferite dacă ordinea arbuștilor diferă. Primele patru soluții obținute sunt, în această ordine: (caprifoi, iasomie, liliac, tamarix, scumpie), (caprifoi, iasomie, liliac, scumpie, tamarix), (caprifoi, iasomie, tamarix, liliac, scumpie), (caprifoi, iasomie, tamarix, scumpie, liliac). Indicați penultima soluție generată.
a. (scumpie, tamarix, caprifoi, iasomie, liliac)
b. (scumpie, tamarix, caprifoi, liliac, iasomie)
c. (scumpie, tamarix, liliac, caprifoi, iasomie)
d. (scumpie, tamarix, liliac, iasomie, caprifoi)
- Fiecare dintre variabilele **A** și **B**, declarate alăturat, memorează coordonatele (x abscisa, iar y ordonata) câte unui punct în sistemul de coordonate xOy . Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă segmentul cu extremitățile în punctele corespunzătoare variabilelor **A** și **B** intersectează axa Oy a sistemului de coordonate.

	<pre>struct punct { int x,y; } A,B;</pre>
--	---

a. $A.x * B.x \leq 0$ b. $A(x) * B(x) \leq 0$ c. $x.A * x.B \leq 0$ d. $\text{punct}(A,B).y == 0$
- Într-un arbore cu rădăcină un nod se află pe nivelul x dacă lanțul elementar care are o extremitate în nodul respectiv și cealaltă extremitate în rădăcina arborelui are lungimea x . Pe nivelul 0 se află un singur nod (rădăcina).
Un arbore cu rădăcină are 8 noduri, numerotate de la 1 la 8, și muchiile [1,3], [1,7], [1,8], [2,4], [3,5], [3,6], [4,5]. Știind că rădăcina arborelui este nodul numerotat cu 5, indicați nodurile situate pe nivelul 2 al arborelui dat.
a. 7,8 b. 7,4 c. 1,4,6 d. 1,2,6
- Un graf orientat cu 5 vârfuri, numerotate de la 1 la 5, are arcele (2,5), (3,1), (5,3), (5,4). Indicați numărul minim de arce care trebuie adăugate acestuia, astfel încât graful obținut să fie tare conex.
a. 1 b. 2 c. 3 d. 4

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

- a. Scrieți numărul afișat în urma executării algoritmului dacă pentru n se citește valoarea 7. **(6p.)**
- b. Scrieți două numere din intervalul $[10, 10^2)$ care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 10. **(6p.)**
- c. Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat una dintre structurile `cât timp...execută` cu o structură repetitivă de tip `pentru...execută`. **(6p.)**

```
citește n (număr natural nenul)
nr ← 0; i ← 1
cât timp i ≤ n execută
  x ← 0; y ← 1; j ← 1
  cât timp j < i execută
    r ← 2*x - y; x ← y; y ← r
    j ← j + 1
  i ← i + 1
  dacă y > 0 atunci
    nr ← nr + 1
scrie nr
```

2. Subprogramul f este definit alăturat. Scrieți două numere naturale din intervalul deschis $[1, 10]$, care pot fi memorate în variabilele întregi x_1 , respectiv x_2 , astfel încât valoarea lui $f(10, x_1)$ să fie 5, iar valoarea lui $f(x_2, 10)$ să fie 1. **(6p.)**

```
int f(int a, int b)
{ if(a > b) return a/b + f(a-b, b);
  if(a < b) return b/a + f(a, b-a);
  return 1;
}
```

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 4 linii și 5 coloane, numerotate începând de la 0, cu elemente numere întregi, inițial toate nule. Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni astfel încât, în urma executării acesteia, variabila a să memoreze tabloul alăturat. **(6p.)**

```
16 17 18 19 20
11 12 13 14 15
 6  7  8  9 10
 1  2  3  4  5
```

SUBIECTUL al III-lea

(30 de puncte)

1. Subprogramul `prodprim` are doi parametri:

- n , prin care primește un număr natural ($n \in [2, 10^9]$);
- p , prin care furnizează produsul divizorilor primi ai lui n .

Scrieți definiția completă a subprogramului.

Exemplu: dacă $n=2000$, în urma apelului $p=10$, deoarece $2000=2^4 \cdot 5^3$.

(10p.)

2. Într-un text cu cel mult 100 de caractere, cuvintele sunt formate din litere 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 menționat și afișează pe ecran, pe linii separate, toate cuvintele sale pentru care numărul de vocale este strict mai mic decât numărul de consoane. Dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele din mulțimea a, e, i, o, u .

Exemplu: pentru textul `ei au plantat tamarix ea a adus iasomie` se afișează pe ecran, nu neapărat în această ordine, cuvintele alăturate.

(10p.) plantat
tamarix

3. Se citesc de la tastatură două numere naturale din intervalul $[1, 81]$, p_1 și p_2 , și se cere scrierea în fișierul `bac.out` a tuturor numerelor naturale cu exact 7 cifre, pentru care produsul primelor două cifre este egal cu p_1 , cele trei cifre din mijloc sunt egale între ele, iar produsul ultimelor două cifre este egal cu p_2 . Numerele apar în fișier în ordine strict descrescătoare, fiecare pe câte o linie. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă $p_1=12$, iar $p_2=8$, atunci 2633324 și 3400018 sunt două dintre cele 160 de numere cu proprietatea cerută ($2 \cdot 6=3 \cdot 4=12$ și $2 \cdot 4=1 \cdot 8=8$).

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 6

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

SUBIECTUL al II - lea

(40 de puncte)

1.	a) Răspuns corect: 4	6p.	
	b) Răspuns corect: 19 20	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței.
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	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.	Răspuns corect: $x_1=6$, $x_2=10$	6p.	Se acordă câte 3p. pentru fiecare valoare conform cerinței.
3.	Pentru rezolvare corectă -acces la un element al tabloului -determinare a valorilor conform cerinței (*) -corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect specific al datelor determinate (valori consecutive pe fiecare linie, valori suport) conform cerinței.

SUBIECTUL al III - lea

(30 de puncte)

1.	Pentru subprogram corect -antet subprogram (*) -determinare a valorii cerute (**) -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului ¹⁾	10p. 3p. 6p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri de intrare, parametri de ieșire) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unui divizor prim, divizori suport, algoritm principial corect de determinare a unui produs).
----	---	---------------------------	---

2.	<p>Pentru program corect -declarare a unei variabile care să memoreze șir de caractere -citire a datelor -determinare a cuvintelor cerute (*) -afișare a datelor în formatul cerut și tratare a cazului nu exista -declarare a variabilelor simple, corectitudine globală a programului¹⁾</p>	<p>10p. 1p. 1p. 6p. 1p. 1p.</p>	<p>(*) Se acordă câte 2p. pentru fiecare aspect specific (identificare a unui cuvânt, identificare a unei vocale/consoane, algoritm de numărare principal corect) conform cerinței.</p>
3.	<p>a) Pentru program corect -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 programului¹⁾</p>	<p>8p. 1p. 5p. 1p. 1p.</p>	<p>(*) 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 eficient, care nu verifică inutil multe valori.</p>
	<p>b) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență</p>	<p>2p. 1p. 1p.</p>	<p>O soluție posibilă generează numerele cerute stabilind toate valorile posibile pentru prima cifră, c1, pentru a treia cifră, c3, respectiv pentru a 6-a cifră, c6, celelalte calculându-se direct pe baza celorlalte, având în vedere ordinea cerută și încadrarea fiecărei cifre în intervalul [0,9], adaptat după caz.</p>

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