

Proba E. d)
INFORMATICĂ
Limbajul C/C++

Varianta 5

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

- 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.
- 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. Indicați o expresie C/C++ cu valoarea 1.

- a. `!(2>3 || 3>4)` b. `!(2>3) && 3>4`
c. `2>3 && !(3>4)` d. `4>3>2`

2. Subprogramul `f` este definit alăturat. Indicați valoarea lui `f(56579)`.

```
int f(int n)
{ if(n==0) return 0;
  if(n%2==1) return f(n/10)+1;
  return f(n/10)-1;
}
```

- a. 0 b. 1 c. 2 d. 3

3. În muzica clasică, o sonată este o lucrare compusă din trei părți. Variabila declarată alăturat memorează indicativul unei sonate și durata fiecăreia dintre cele trei părți ale acesteia, exprimată în minute și secunde. Indicați o expresie a cărei valoare este egală cu durata celei de a doua părți a sonatei, exprimată în secunde.

```
struct sonata
{ int indicativ;
  struct
  { int min, sec;
  } unu, doi, trei;
} s;
```

- a. `s.doi.sec+60*s.doi.min` b. `sonata.doi.sec+60*sonata.doi.min`
c. `sec.doi.s+60*min.doi.s` d. `doi.sec.sonata.s+60*doi.min.sonata.s`

4. Într-un arbore cu rădăcină considerăm că 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 8 noduri, numerotate de la 1 la 8, este reprezentat prin vectorul de tați: (3, 4, 0, 3, 4, 8, 2, 3). Indicați numărul de niveluri ale arborelui.

- a. 7 b. 5 c. 4 d. 2

5. Un graf orientat cu 7 vârfuri, numerotate de la 1 la 7, are arcele (1, 2), (2, 3), (3, 1), (4, 5), (4, 6), (5, 6), (6, 7), (7, 4). Indicați numărul minim de arce care pot fi adăugate și poziționate adecvat, astfel încât graful orientat 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.

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 valoarea afișată în urma executării algoritmului dacă se citesc, în această ordine, numerele 120, 126, 2. (6p.)
- b. Dacă pentru `m` se citește numărul 2026, iar pentru `k` se citește numărul 6, scrieți două numere din intervalul $[10^3, 10^4]$ care pot fi citite pentru `n`, astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze valoarea 2. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura `pentru...execută` cu o structură de tip `cât timp...execută`. (6p.)

```
citește m,n,k
(numere naturale nenule,
 m≤n, 0≤k≤9)
nr←0
pentru i←m,n execută
  x←i
  repetă
    dacă x%10=k atunci
      nr←nr+1
  x←[x/10]
până când x=0
scrie nr
```

2. Lacătul unei biciclete are un cifru format din 5 cifre distincte, astfel încât oricare două cifre alăturate au parități diferite. Pentru că a uitat cifrul, Andrei încearcă toate soluțiile posibile, pe care le generează în ordine lexicografică, utilizând metoda backtracking. Două cifruri diferă prin cel puțin o cifră sau prin ordinea acestora, primele șase fiind, în ordinea generării: **01234, 01236, 01238, 01254, 01256, 01258**. Scrieți două soluții, una generată imediat înainte, iar cealaltă generată imediat după **16789**. (6p.)
3. Variabilele **s** și **t** permit accesul la câte un șir de cel mult 50 de caractere; inițial **t** accesează un șir vid, iar **s** un șir ce memorează, în această ordine, numele și prenumele unui profesor, separate printr-un spațiu și formate din litere mari și mici ale alfabetului limbii engleze. Scrieți o secvență de instrucțiuni C/C++ în urma executării căreia variabila **t** accesează șirul format, în această ordine, din titlu (**Prof.**), urmat de prenume și nume, separate prin câte un spațiu, ca în exemplu. Declarați eventuale alte variabile necesare.
Exemplu: dacă **s** accesează șirul **POP Grigore**, atunci **t** accesează șirul **Prof. Grigore POP** (6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. O editură realizează seturi turistice, formate din câte o broșură de prezentare a zonei locale și câte un catalog al producătorilor locali, astfel încât dacă o broșură are **p** pagini, atunci **p** este un număr prim, iar catalogul din același set are **3 · p** pagini.
Subprogramul **pagini** are un parametru, **n**, prin care primește un număr natural ($n \in [2, 10^4]$). Subprogramul returnează valoarea 1 dacă, folosind toate cele **n** pagini, se poate realiza un set turistic, sau valoarea 0 în caz contrar. Scrieți în C/C++ definiția completă a subprogramului.
Exemplu: dacă **n=44**, subprogramul returnează valoarea 1 (se poate obține o broșură de 11 pagini și un catalog de $33=3 \cdot 11$ pagini), iar dacă **n=48** sau **n=27**, subprogramul returnează 0. (10p.)
2. Într-un tablou bidimensional, două coloane sunt numite **stabile** dacă suma valorilor de pe una dintre ele este egală cu suma valorilor de pe cealaltă, iar cele două coloane sunt consecutive în tablou.
Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul $[1, 20]$, **m** și **n**, și **m · n** numere naturale din intervalul $[0, 10]$, elemente ale unui tablou bidimensional cu **m** linii și **n** coloane, numerotate începând de la 0. Programul afișează pe ecran toate perechile de numere **x** și **y** (**x < y**), pentru care coloanele **x** și **y** sunt stabile în tabloul dat, sau mesajul **nu exista**, dacă nu există astfel de coloane.
Fiecare pereche se afișează pe câte o linie a ecranului, cu cele două valori în ordine strict crescătoare, separate prin câte un spațiu.
Exemplu: pentru **m=6**, **n=7** și tabloul alăturat, se afișează pe ecran perechile de mai jos, nu neapărat în această ordine
0 1
1 2
5 6
(10p.)
- | | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 6 | 1 | 0 | 4 | 0 | 0 |
| 0 | 3 | 5 | 2 | 3 | 1 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 |
| 6 | 0 | 5 | 0 | 6 | 0 | 1 |
| 1 | 3 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 2 | 0 | 0 |
3. Șirul 1, 2, 2, 5, 4, 11, 8, 23, 16, 47 . . .
este definit astfel: $f_0=1$, iar pentru orice **n**, număr natural strict pozitiv, $f_n=3 \cdot f_{n-1} - 1$, dacă **n** este impar, sau $f_n=2 \cdot f_{n-2}$, dacă **n** este par.
Se citește de la tastatură un număr natural **x** ($x \in [4, 10^9]$), termen al șirului dat. Se cere să se scrie în fișierul **bac.txt**, în ordinea inversă apariției în șir, separați prin câte un spațiu, atât termenul citit, cât și toți cei care îl preced în șir pe acesta, ca în exemplu. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.
Exemplu: dacă **x=8**, fișierul conține numerele 8 11 4 5 2 2 1
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)