

Examenul de bacalaureat național 2020

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 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. Variabila x este de tip întreg și poate memora un număr natural din intervalul $[0, 10^9)$. Indicați valoarea maximă pe care o poate avea expresia C/C++ alăturată. $x \times 2020$
- a. 20.2 b. 2019 c. 2020 d. 495049
2. Subprogramul f este definit alăturat. Indicați apelul în urma căruia simbolul $*$ se afișează de trei ori.
- ```
void f(int x)
{
 cout<<"*"; printf("*");
 if(x>5) f((x+1)/2);
}
```
- a.  $f(30)$ ;                      b.  $f(21)$ ;                      c.  $f(17)$ ;                      d.  $f(8)$ ;
3. Utilizând metoda backtracking se generează toate modalitățile de a prepara o băutură din câte 3 sau 4 tipuri de fructe distincte din mulțimea {căpșune, mere, pepene, pere, portocale}. Primele patru soluții generate sunt, în această ordine: (căpșune, mere, pepene), (căpșune, mere, pepene, pere), (căpșune, mere, pepene, portocale), (căpșune, mere, pere). Indicați cea de a șasea soluție generată.
- a. (pepene, pere, portocale)                      b. (mere, pepene, pere, portocale)  
c. (căpșune, mere, portocale)                      d. (căpșune, mere, pere, pepene)
4. Un arbore cu 9 noduri, numerotate de la 1 la 9, este reprezentat prin vectorul de „tați” (8, 6, 4, 2, 3, 0, 4, 2, 7). Indicați un „frate” al nodului 4.
- a. 1                      b. 2                      c. 7                      d. 8
5. Un graf neorientat eulerian cu 5 noduri are un nod cu gradul 4. Indicați numărul minim de muchii ale grafului.
- a. 4                      b. 6                      c. 8                      d. 10

**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  $c$ .
- ```
citește n (număr natural)
m ← 0
repetă
    c ← n % 10; n ← [n/10]
    dacă c < 5 atunci m ← m - 2 * c
    altfel m ← m + c
    ■
până când n = 0
dacă m = 0 atunci scrie 'DA'
altfel scrie m, 'NU'
■
```
- a. Scrieți ce se afișează dacă se citește numărul 247388. (6p.)
- b. Scrieți două numere din intervalul $[10^2, 10^3)$, cu cifre distincte, care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze DA. (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 cu test inițial. (6p.)

2. Variabila **p** memorează simultan, pentru fiecare dintre cele **20** de tipuri de procesoare dintr-un depozit, următoarele date: producătorul (o literă mare a alfabetului englez), frecvența procesorului (număr natural) și prețul produsului (număr real). Știind că expresiile C/C++ de mai jos au ca valori producătorul primului procesor, respectiv frecvența și prețul acestuia, scrieți definiția unei structuri cu eticheta **procesor**, care permite memorarea datelor despre un procesor, și declarați corespunzător variabila **p**.

p[0].producator p[0].frecventa p[0].pret (6p.)

3. Variabilele **i**, **j** și **k** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu **6** linii și **4** coloane, numerotate începând de la **0**, cu elemente numere naturale din intervalul **[0, 50]**. Scrieți o secvență de instrucțiuni în urma executării căreia să se rearanjeze elementele pare situate pe coloana a **3**-a a tabloului, astfel încât acestea să fie ordonate crescător, de sus în jos, în cadrul aceleiași coloane, iar celelalte elemente ale tabloului să își păstreze pozițiile, ca în exemplu.

Exemplu: pentru tabloul

2	3	4	5
8	7	8	5
1	3	5	7
3	0	2	9
3	5	1	6
7	3	0	2

se obține tabloul

2	3	0	5
8	7	2	5
1	3	5	7
3	0	4	9
3	5	1	6
7	3	8	2

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Un număr natural nenul se numește **p-număr** dacă are aceeași paritate cu suma divizorilor săi pozitivi.
Exemplu: **10** și **25** sunt p-numere (**10** are aceeași paritate cu **18=1+2+5+10**, iar **25** are aceeași paritate cu **31=1+5+25**).

Subprogramul **kpn**, are trei parametri, **a**, **b** și **k**, prin care primește câte un număr natural din intervalul **[1, 10⁶]** (**a ≤ b**). Subprogramul returnează cel de al **k**-lea p-număr din intervalul **[a, b]** sau **-1**, dacă nu există cel puțin **k** astfel de numere în acest interval. Scrieți definiția completă a subprogramului.

Exemplu: dacă **a=27**, **b=50** și **k=3**, atunci subprogramul returnează **34**. **(10p.)**

2. Două cuvinte distincte se numesc în **oglină** dacă fiecare dintre ele se obține prin citirea literelor celuilalt de la dreapta la stânga.

Exemplu: **animate** și **etamina** sunt în oglindă, iar pentru cuvântul **reper** nu există un cuvânt cu care să fie în oglindă.

Se consideră un text cu cel mult **100** de caractere, în care 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 mai sus și îl transformă în memorie, înlocuind fiecare cuvânt cu număr impar de litere cu acel cuvânt cu care el este în oglindă, dacă acesta există, ca în exemplu. Programul afișează pe ecran textul obținut sau mesajul **nu exista**, dacă în text nu s-a înlocuit niciun cuvânt.

Exemplu: pentru textul era o selectie reper de desene animate prezenta

se obține textul are o selectie reper de desene etamina prezenta

iar pentru textul un reper pentru desene

se afișează pe ecran mesajul **nu exista** **(10p.)**

3. Fișierul **bac.txt** conține un șir de cel mult **10⁶** numere naturale din intervalul **[0, 10⁹]**, separate prin câte un spațiu.

Se cere să se afișeze pe ecran, separate printr-un spațiu, două numere naturale **a** și **b** (**a < b**), astfel încât oricare termen al șirului care are exact două cifre să aparțină intervalului **(a, b)**, iar valoarea expresiei **b-a** să fie minimă. Dacă șirul nu are niciun termen de două cifre, pe ecran se afișează mesajul **nu exista**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu: dacă fișierul conține valorile **7 2 40 5 11 15 10 122 18 350**

se afișează pe ecran numerele **9 41**.

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

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