

• Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu. • Timpul efectiv de lucru este de 3 ore. • În rezolvările cerute, identificatorii utilizați 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). • În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

SUBIECTUL I**(20 de puncte)**

1. Indicați expresia care are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă x are exact două cifre, iar cifra unităților este nenulă.

a. $(x/10)*(x\%10)!=0 \ \&\& \ x/100==0$

b. $(x/10)*(x\%10)!=0 \ \&\& \ x\%100==0$

c. $(x/10)+(x\%10)!=0 \ || \ x/100==0$

d. $(x/10)+(x\%10)!=0 \ || \ x\%100==0$

2. Un arbore are 4 frunze, iar lungimea oricărui lanț elementar care unește două noduri de tip frunză este egală cu 6. Numărul minim de noduri ale unui astfel de arbore este:

a. 6

b. 9

c. 13

d. 25

3. Utilizând metoda backtracking se generează, în ordine lexicografică, toate șirurile de maximum 4 litere distincte din mulțimea $\{a, v, i, o, n\}$. Primele 5 șiruri generate sunt, în această ordine: a, ai, ain, aino, ainv. Imediat după șirul avn se generează:

a. avi

b. avni

c. avo

d. iano

4. Se consideră subprogramele $f1$ și $f2$, definite mai jos.

long f1 (int n)

```
{
if(n<10) return n*f1(n+1);
return 10;
}
```

long f2 (int n)

```
{ if(n>1) return n*f2(n-1);
return 1; }
```

Identificați subprogramul care, la apel, pentru parametrul $n=10$, returnează produsul primelor 10 numere naturale nenule.

a. atât $f1$, cât și $f2$ b. numai $f1$ c. numai $f2$ d. nici $f1$, nici $f2$

5. Se consideră un graf neorientat conex și fără cicluri, în care gradul oricărui nod este mai mic sau egal cu 4. Dacă șase dintre nodurile sale au gradul egal cu 1, atunci numărul maxim de noduri cu gradul egal cu 4 este:

a. 0

b. 1

c. 2

d. 3

SUBIECTUL II**(40 de puncte)**

1. Se consideră algoritmul alăturat, reprezentat în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întregă a numărului real z .

citește n (număr natural nenul)

 $m \leftarrow n$ $p \leftarrow 1$

┌cât timp $n \geq p * 10$ execută

| $c1 \leftarrow [n/p]\%10$ | $c2 \leftarrow [n/(p*10)]\%10$ | ┌dacă $c1 > c2$ atunci|| $n \leftarrow n - (c1 - c2) * p$ || $p \leftarrow p * 10$ || $n \leftarrow n + (c1 - c2) * p$

| └altfel

|| $p \leftarrow p * 10$

| └

└

┌dacă $n = m$ atunci

| scrie 0

| └altfel

| scrie n

└

a) Scrieți numărul afișat dacă se citește valoarea 162453. (6p.)

b) Scrieți două numere de patru cifre distincte care pot fi citite astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze valoarea 0. (6p.)

c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura cât timp...execută cu o structură repetitivă cu test final. (6p.)

2. Se consideră declarațiile alăturate, în care variabila s memorează coordonatele (x abscisa, iar y ordonata), în sistemul de coordonate xOy, ale extremităților unui segment.

struct punct

```
{ int x,y; };
```

struct segment

```
{ punct A,B; }s;
```

Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă ambele extremități ale segmentului aparțin axei Ox a sistemului de coordonate. (6p.)

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg, iar variabila A memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, cu elemente numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos.

```
for(i=1;i<=5;i++)
```

```
3 4 5 6 7
```

```
for(j=1;j<=5;j++)
```

```
5 6 7 8 9
```

```
..... (6p.)
```

```
7 8 9 10 11
```

```
9 10 11 12 13
```

```
11 12 13 14 15
```

SUBIECTUL III

(30 de puncte)

1. Subprogramul **număr** are doi parametri:

- n, prin care primește un număr natural ($2 \leq n \leq 10^9$)

- m prin care furnizează un număr natural ($2 \leq m \leq 10^9$) cel mai mare număr natural care se poate obține folosind toate cifrele impare care apar în scrierea lui n, dacă în scrierea lui n nu apar cifre impare atunci m va avea valoarea -1. Scrieți în limbajul C/C++ definiția completă a subprogramului **număr**. **Exemplu: dacă n=74317 atunci m=7731, iar dacă n=246 atunci m=-1.**

2. Numim **dublu-cuvânt** un cuvânt cu număr par de litere care este format prin dublarea unui cuvânt. De exemplu, **tictic** este un dublu-cuvânt, în timp ce **tictac** sau **abba** nu sunt. Scrieți un program C/C++ care citește de la tastatură un text de lungime maximă 255 de caractere, format doar din cuvinte separate prin câte un spațiu și modifică în memorie șirul înlocuind a doua jumătate a fiecărui dublu-cuvânt cu caracterul *. Dacă s-a făcut cel puțin o modificare se va afișa șirul rezultat, altfel se va afișa mesajul **NEMODIFICAT**.

Exemplu: dacă se citește : **se aude tictic tictac apoi dingding dingding** se va afișa: **se aude tic* tictac apoi ding* ding***

3. Fișierul BAC.TXT conține pe prima linie un număr natural, n ($n \in [2, 5000]$), și pe a doua linie un șir de $2 \cdot n$ numere naturale din intervalul [0,5]. Numerele aflate pe aceeași linie a fișierului sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran valoarea obținută însumând toate produsele de forma $x \cdot y$, unde x și y sunt numere de paritate diferită, x fiind printre primii n termeni ai șirului aflat în fișier, iar y printre ultimii n termeni ai acestui șir. Dacă nu există niciun astfel de produs, valoarea cerută este nulă. Pentru determinarea numărului cerut utilizați un algoritm eficient din punctul de vedere al timpului de executare și al memoriei necesare. Exemplu: dacă fișierul are conținutul de mai jos 5 1 2 0 0 5 4 2 3 1 0 pe ecran se afișează numărul 44 ($1 \cdot 4 + 1 \cdot 2 + 1 \cdot 0 + 2 \cdot 3 + 2 \cdot 1 + 0 \cdot 3 + 0 \cdot 1 + 0 \cdot 3 + 0 \cdot 1 + 5 \cdot 4 + 5 \cdot 2 + 5 \cdot 0 = 44$).

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

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