

## Solutii propuse pentru exercitiile din sesiunea 2

---

1. Scrieti un subprogram care elimina toate cifrele impare dintr-un numar **n** primit ca parametru. Se garanteaza faptul ca **n**  $\geq 10$ ; Programul va return numarul modificat.

- Exemplu: Pentru **n** = 123456, subprogramul va intoarce numarul: 246
- Solutie:

```
#include <iostream>
using namespace std;

int eliminaImpare(int n);

int main() {

    int n = 123456;

    cout << eliminaImpare(n);

    return 0;
}

int eliminaImpare(int n) {
    int rezultat = 0;
    int pozitie = 1;
    while (n > 0) {
        int ultimaCifra = n % 10;
        if (ultimaCifra % 2 == 0) {
            rezultat = ultimaCifra * pozitie + rezultat;
            pozitie = pozitie * 10;
        }
        n = n/10;
    }

    return rezultat;
}
```

2. Scrieti un subprogram care primeste ca si parametru un numar **n**. Subprogramul va inlocui fiecare cifra impara cu dublul acesteia. In cazul in care dublul cifrei impare va fi mai mare decat 10, se va lua ultima cifra.

- Exemplu pentru **n** = 123456, subprogramul va intoarce numarul: 226406
- Solutie:

```
#include <iostream>
using namespace std;

int dubleazaImpare(int n);
```

```
int main() {  
  
    int n = 123456;  
  
    cout << dubleazaImpare(n);  
  
    return 0;  
}  
  
int dubleazaImpare(int n) {  
    int rezultat = 0;  
    int pozitie = 1;  
    while (n > 0) {  
        int ultimaCifra = n % 10;  
        if (ultimaCifra % 2 == 1) {  
            ultimaCifra = (ultimaCifra * 2) % 10;  
        }  
  
        rezultat = ultimaCifra * pozitie + rezultat;  
        pozitie = pozitie * 10;  
        n = n/10;  
    }  
  
    return rezultat;  
}
```

3. Un joc online cu  $n$  jetoane poate fi jucat de un grup de  $k$  ( $k \geq 2$ ) jucători, numai dacă toate cele  $n$  jetoane pot fi distribuite în mod egal celor  $k$  jucători. Subprogramul `joc` are un singur parametru,  $n$ , prin care primește un număr natural ( $n$  aparține lui  $[2, 104]$ ), reprezentând numărul de jetoane ale unui joc de tipul precizat. Subprogramul returnează numărul valorilor distincte pe care le poate avea  $k$  pentru acest joc. Scrieți definiția completă a subprogramului. Exemplu: dacă  $n=12$ , atunci subprogramul returnează numărul 5 (cele 12 jetoane se pot distribui în mod egal pentru o grupă de 2 jucători, de 3 jucători, de 4 jucători, de 6 jucători sau de 12 jucători)

- Link: <https://modinfo.ro/bac/variante-test-2021/info/v4.pdf>

- Soluție:

```
#include <iostream>  
using namespace std;  
  
int impartireJetoane(int n);  
  
int main() {  
  
    int n = 12;  
  
    cout << impartireJetoane(n);  
  
    return 0;  
}
```

```
int impartireJetoane(int n) {
    int contor = 0;
    for (int i = 2; i <= n; i++) {
        if (n % i == 0) {
            contor++;
        }
    }

    return contor;
}
```

4. Subprogramul identice are un singur parametru, n, prin care primește un număr natural (n aparține intervalului [10,109]). Subprogramul returnează valoarea 1, dacă numărul n are toate cifrele egale, sau valoarea 0 în caz contrar. Scrieți definiția completă a subprogramului. Exemplu: dacă n=2222, subprogramul returnează valoarea 1, iar dacă n=212, subprogramul returnează valoarea 0

- Link: <https://modinfo.ro/bac/variante-test-2021/info/v5.pdf>

- Solutie:

```
#include <iostream>
using namespace std;

int identice(int n);

int main() {
    int n = 22221;

    cout << identice(n);

    return 0;
}

int identice(int n) {
    int egale = 1;
    int ultimaCifraInitiala = n % 10;
    n = n/10;
    while (n > 0) {
        int ultimaCifraCurenta = n % 10;
        if (ultimaCifraInitiala != ultimaCifraCurenta) {
            egale = 0;
            break;
        }
        n = n/10;
    }

    return egale;
}
```

5. Subprogramul **afisare** are trei parametri:

- $x$  și  $y$ , prin care primește câte un număr natural din intervalul  $[0,106]$  ( $x \leq y$ );
- $k$ , prin care primește un număr natural ( $k \in [2,102]$ ).

Subprogramul afișează pe ecran, în ordine strict crescătoare, numerele din intervalul  $[x,y]$ , în secvențe de câte  $k$ , cu excepția ultimei secvențe care poate conține mai puțin de  $k$  numere.

- Fiecare secvență se încheie cu câte un simbol **\***, iar numerele și simbolurile sunt separate prin câte un spațiu, ca în exemplu. Scrieți definiția completă a subprogramului. Exemplu: dacă  $x=11$ ,  $y=21$  și  $k=4$  se afișează pe ecran numerele de mai jos, în acest format. 11 12 13 14 \* 15 16 17 18 \* 19 20 21 \*
- Soluție:

```
#include <iostream>
using namespace std;

void afisare(int x, int y, int k);

int main() {
    int x = 11, y = 21, k = 4;

    afisare(x,y,k);

    return 0;
}

void afisare(int x, int y, int k) {
    int contorNumereAfisate = 1;
    for (int i = x; i <= y; i++) {
        cout << i << " ";
        if ( contorNumereAfisate == k || i == y) {
            cout << "* ";
            contorNumereAfisate = 0;
        }
        contorNumereAfisate++;
    }
}
```

6. Să se scrie un subprogram C++ prin care se dublează prima cifră a unui număr natural  $n$  transmis ca parametru. Funcția întoarce rezultatul prin intermediul aceluiași parametru  $n$ .

- Link:
  - Problema este luata de aici: <https://www.pbinfo.ro/probleme/1633/dublare1>. Poti vedea la pagina asta mai multe detalii, cum ar fi un exemplu de date de intrare dar desigur, te poti si verifica la ei pe site.
- Soluție:

```
#include <iostream>
using namespace std;

void dublare1(int &n);
int main() {

    int n = 85412;
    dublare1(n);
    cout << n;
    return 0;
}

void dublare1(int &n) {
    int copieN = n;
    int primaCifra;
    int pozitie = 1;
    while(n > 0) {
        primaCifra = n % 10;
        pozitie = pozitie * 10;
        n = n/10;
    }

    n = primaCifra * pozitie + copieN;
}
```

- Solutie 2 (doar de dragul exemplului):

```
#include <cmath>

void dublare1(int &n) {
    int nCopy = n;
    int result = 0;
    int count = 0;
    while (nCopy > 0) {
        count++;
        nCopy /= 10;
    }
    int firstDigit = n / (int)pow(10, count-1);
    n = firstDigit * pow(10, count) + n;
}
```

7. Să se scrie o funcție C++ care primește ca parametri două numere n și k și determină numărul format din primele k cifre ale lui n. Funcția va întoarce rezultatul prin intermediul unui parametru de ieșire.

- Link:
  - Problema este luata de aici: <https://www.pbinfo.ro/probleme/910/kprefix>. Poti vedea la pagina asta mai multe detalii, cum ar fi un exemplu de date de intrare dar desigur, te poti si verifica la ei pe site.

- Solutie:

```
#include <iostream>
using namespace std;

void prefix(int n, int k, int &output);

int main() {

    int n = 27594;
    int k = 3;
    int x;
    prefix(n, k, x);
    cout << x;
    return 0;
}

void prefix(int n, int k, int &output) {
    int rezultat = 0;
    int pozitie = 1;

    while (pozitie * 10 <= n) {
        pozitie = pozitie * 10;
    }

    for (int i = 0; i < k; i++) {
        int cifraCurenta = n / pozitie;
        rezultat = rezultat * 10 + cifraCurenta;

        n = n % pozitie;
        pozitie = pozitie / 10;
    }

    output = rezultat;
}
```

8. Subprogramul **numar** are trei parametri:

- **n** și **c**, prin care primește câte un număr natural (**n** aparține intervalului [0,109], **c** aparține intervalului [0,9]);
- **m**, prin care furnizează numărul obținut din **n**, prin eliminarea din acesta a tuturor cifrelor egale cu **c**, sau -1 dacă toate cifrele lui **n** sunt egale cu **c**. Cifrele nule nesemnificative sunt ignorate, ca în exemplu. Scrieți definiția completă a subprogramului. Exemplu: dacă **n=50752** sau **n=72** și **c=5**, după apel **m=72**, dacă **n=500** și **c=5**, după apel **m=0**, iar dacă **n=55** și **c=5**, după apel **m=-1**.
- Solutie:

```
#include <iostream>
using namespace std;
```

```
void numar(int n, int c, int &m);

int main() {

    int n = 5500;
    int c = 5;
    int m;
    numar(n, c, m);
    cout << m;
    return 0;
}

void numar(int n, int c, int &m) {
    int rezultat =0;
    int areDiferiteDeC =0;
    int pozitie = 1;
    while (n > 0) {
        int ultimaCifra = n % 10;
        if (ultimaCifra != c) {
            rezultat = ultimaCifra * pozitie + rezultat;
            pozitie = pozitie * 10;
            if (ultimaCifra != 0) {
                areDiferiteDeC = 1;
            }
        }
        n = n / 10;
    }

    if (!areDiferiteDeC) {
        m = -1;
    } else {
        m = rezultat;
    }
}
```

9. Să se scrie o funcție C++ recursivă care să returneze suma cifrelor unui număr natural transmis ca parametru.
- Link: <https://www.pbinfo.ro/probleme/823/sumcifrec> Poti vedea la pagina asta mai multe detalii, cum ar fi un exemplu de date de intrare dar desigur, te poti si verifica la ei pe site.
  - Solutie:

```
#include <iostream>
using namespace std;

int suma(int n);

int main() {
```

```
    int n = 12345;
    cout << suma(n);

    return 0;
}

int suma(int n) {
    if (n == 0) {
        return 0;
    } else {
        int ultimaCifra = n % 10;
        return ultimaCifra + suma(n/10);
    }
}
```

10. Să se scrie o funcție C++ recursivă care determină cel mai mare divizor comun a două numere transmise ca parametri și întoarce rezultatul prin intermediul unui parametru de ieșire.

- Link: <https://www.pbinfo.ro/probleme/917/cmmdcrec1> Poti vedea la pagina asta mai multe detalii, cum ar fi un exemplu de date de intrare dar desigur, te poti si verifica la ei pe site.
- De asemenea, problema asta e mai tricky putin, deoarece combina doua concepte pe care le-am invatat in sesiunea asta, anume functiile recursive si functiile care intorc valori prin intermediul parametrilor de intrare.
- Solutie:

```
#include <iostream>
using namespace std;

void cmmdc(int a, int b, int& r);

int main() {

    int a = 24;
    int b = 18;
    int rezultat;
    cmmdc(a, b, rezultat);

    cout << rezultat;

    return 0;
}

void cmmdc(int a, int b, int& r) {
    if (b == 0){
        r = a;
        return;
    }
    int reminder = a % b;
```



```
    cmmdc(b, reminder, r);  
}
```