



Instrucțiunea WHILE – Probleme cu cifrele unui număr

Forma generală

```
while (expresie)
    instrucțiuni;
```

Semnificație

Cât timp expresia este adevărată, se execută setul de instrucțiuni. **Atentie!** În bucla *while* trebuie să existe o instrucțiune care asigură terminarea iterațiilor (repetițiilor).

Probleme rezolvate

1. Suma cifrelor unui număr natural **n**, de cel mult 9 cifre.

```
#include <iostream>
using namespace std;
int main()
{
    int n, s=0;
    cin>>n;
    while(n)
    {
        s=s+n%10;
        n=n/10;
    }
    cout<<s;
    return 0;
}
```

2. Produsul cifrelor impare ale unui număr natural **n**, de cel mult 9 cifre. Dacă numărul nu conține nicio cifră impară, se va afișa -1.

```
#include <iostream>
using namespace std;
int main()
```



```
{
    int n,p=1,nrcifre=0;
    cin>>n;
    while(n)
    {
        if(n%2)
        {
            nrcifre++;
            p=p*(n%10);
        }
        n=n/10;
    }
    if(nrcifre==0)
        cout<<-1;
    else
        cout<<p;
    return 0;
}
```

3. Se citesc două numere naturale, **n** – un număr natural de cel mult 9 cifre și **c** – o cifră din mulțimea {0,1,2,... 9}. Afișați de câte ori apare **c** în scrierea numărului **n**.

```
#include <iostream>
using namespace std;
int main()
{
    int n,c,nrcifre=0;
    cin>>n>>c;
    if(n==0 and c==0)
        nrcifre=1;
    while(n)
    {
        if(n%10==c)
        {
            nrcifre++;
        }
        n=n/10;
    }
    cout<<nrcifre;
    return 0;
}
```



4. Se citește un număr natural **n** cu cel mult 9 cifre. Apoi se introduc de la tastatură două valori **c1** și **c2** reprezentând două cifre (**c2** este totdeauna o cifră nenulă!). Afișați numărul **n** după înlocuirea tuturor cifrelor **c1** cu cifra **c2**.

```
#include <iostream>
using namespace std;
int main()
{
    int n, c1, c2, x=0, p=1;
    cin>>n>>c1>>c2;
    if(n==0 and c1==0)
        n=c2;
    while(n)
    {
        if(n%10==c1)
        {
            x=x+c2*p;
            p=p*10;
        }
        else
        {
            x=x+(n%10)*p;
            p=p*10;
        }
        n=n/10;
    }
    n=x;
    cout<<n;
    return 0;
}
```

5. Se citește un număr natural **n** cu cel mult 9 cifre. Afișați cifra maximă și cifra minimă.

```
#include <iostream>
using namespace std;
int main()
{
    int n, cmax=0, cmin=9;
    cin>>n;
    while(n)
    {
```



```

        if (n%10<cmin)
            cmin=n%10;
        if (n%10>cmax)
            cmax=n%10;
        n=n/10;
    }
    cout<<cmin<<" "<<cmax;
    return 0;
}

```

6. Se citește un număr natural **n** cu cel mult 9 cifre. Afișați cea mai din dreapta cifră pară și cea mai din stânga cifră impară. Dacă **n** nu conține cifre pare se va afișa -1. Dacă **n** nu conține cifre impare se va afișa -2.

Exemplu: Pentru **n=1278** se va afișa: **8 1**.

```

#include <iostream>
using namespace std;
int main()
{
    int n, cpara, cimpara, npare=0, nimpare=0;
    cin>>n;
    if (n==0) {npare=1; cpara=0;}
    while (n)
    {
        if (n%2==0 and npare==0)
        {
            cpara=n%10;
            npare++;
        }
        if (n%2)
        {
            cimpara=n%10;
            nimpare++;
        }
        n=n/10;
    }
    if (npare==0)
        cout<<-1<<" ";
    else
        cout<<cpara<<" ";
    if (nimpare==0)

```



```

        cout<<-2<<" ";
    else
        cout<<cimpara;
    return 0;
}

```

7. Se citește un număr natural **n** cu cel mult 8 cifre. Afișați numărul după efectuarea următoarelor schimbări: se inversează prima cifră cu a doua, a treia cu a patra, etc. Dacă numărul dat are un număr impar de cifre se va afișa mesajul **nu se poate efectua**. Exemplu: **n=1234** se va transforma în **n=2143**.

```

#include <iostream>
using namespace std;
int main()
{
    int n,x=0,p=1,nr;
    cin>>n;
    int cn=n,nrcifre=0;
    while(cn)
    {
        nrcifre++;
        cn/=10;
    }
    if(nrcifre%2!=0)
        cout<<"nu se poate efectua";
    else
    {
        while(n>9)
        {
            nr=n%10*10+n/10%10;
            x=x+nr*p;
            p=p*100;
            n=n/100;
        }
        cout<<x;
    }
    return 0;
}

```

8. Se citește un număr natural **n** cu cel mult 9 cifre. Afișați oglinditul lui **n**. Exemplu: pentru **n=1234** se va afișa **4321**.



```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int cn=n,ogl=0;
    while(cn)
    {
        ogl=ogl*10+cn%10;
        cn/=10;
    }
    cout<<ogl;
    return 0;
}
```

9. Se citește un număr natural **n** cu cel mult 9 cifre. Afișați mesajul **este palindrom** sau **nu este palindrom** dacă numărul are sau nu această proprietate. Un număr este palindrom dacă este egal cu oglinditul său.

Exemplu: pentru **n=1234321** se va afișa **este palindrom**.

Știați că?... Cel mai lung palindrom din lume este un text scris în finlandeză de Teemu Paavolainen în 1992? Acesta are 49 935 de caractere. (*sursa: ro.wikipedia.org*)

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int cn=n,ogl=0;
    while(cn)
    {
        ogl=ogl*10+cn%10;
        cn/=10;
    }
    if(ogl==n)
        cout<<"este palindrom";
    else
        cout<<"nu este palindrom";
}
```



```
    return 0;  
}
```

Probleme propuse

1. Se dau două numere naturale **a**, **b**. Verificați și afișați un mesaj corespunzător dacă numerele sunt formate din aceleași cifre, nu neapărat în aceeași ordine.
2. Se dă un număr natural **n** de maxim 9 cifre. Afișați toate numerele valide care se pot obține din **n** prin permutarea circulară la dreapta a cifrelor acestuia.
3. Se dă un număr natural **n** de maxim 9 cifre. Afișați toate numerele valide care se pot obține din **n** prin permutarea circulară la stânga a cifrelor acestuia.
4. Se citesc două numere naturale nenule **a** și **b**. Afișați, în cazul în care acestea au același număr de cifre, diferența minimă absolută dintre două cifre ce ocupă aceleași poziții în **a**, respectiv în **b** (cifra unităților cu cifra unităților, etc). În caz că nu au număr egal de cifre se va afișa cel mai mare oglindit care se poate obține din **a** sau din **b**.
5. Se dă un număr natural nenul **n** cu maxim 9 cifre. Afișați cel mai mare palindrom care se poate obține din toate cifrele lui **n**. În cazul în care acest lucru nu este posibil, se va afișa -1.
6. Se dă un număr natural nenul **n** cu maxim 9 cifre. Afișați numărul obținut după eliminarea tuturor cifrelor pare.
7. **Cifra de control.** Cifra de control se poate obține astfel: se efectuează suma cifrelor numărului natural **n** dat. Dacă aceasta are mai mult de o cifră, **n** devine suma cifrelor sale și se continuă algoritmul până când **n** ajunge de o cifră. Aceasta este cifra de control. Exemplu: pentru **n=247** se obțin pe rând valorile: 13 (2+4+7), 4 (1+3). Se va afișa **4**.
 - a) Scrieți un program care determină cifra de control pentru un număr natural dat;
 - b) Scrieți programul pentru determinarea cifrei de control folosind o singură structură repetitivă;
 - c) Scrieți programul pentru determinarea cifrei de control fără nicio structură repetitivă.

