

# Рекурзија и рекурзивни функции

Низата на непарни броеви 1, 3, 5, 7,... започнува со 1 и секој следен се добива кога на претходниот му се додава 2. Ако  $x_n$  е  $n$ -тиот непарен број, тогаш  $(n+1)$ -тиот непарен број е  $x_{n+1} = x_n + 2$ .

Ваквите релации каде што следниот елемент се пресметува од еден или повеќе претходно дадени елементи се викаат **рекурентни релации**.

Пр. Факториел на природен број

$$n! = n * (n-1) * (n-2) * \dots * 1 \quad n \geq 1, 0! = 1$$

$$\text{За } n=1, \quad 1! = 1 * 0! = 1$$

$$\text{За } n=2, \quad 2! = 2 * (2-1)! = 2 * 1! = 2 * 1 = 2 \text{ и.т.н.}$$

Пр. Фибоначиеви броеви

$$F_n = F_{n-1} + F_{n-2}, \text{ за } n=0, 1, 2, \dots \text{ и почетни } F_0=1, F_1=1$$

$$\text{за } n=2, \quad F_2 = F_1 + F_0 = 1 + 1 = 2$$

$$\text{за } n=3, \quad F_3 = F_2 + F_1 = 2 + 1 = 3$$

$$\text{за } n=4, \quad F_4 = F_3 + F_2 = 3 + 2 = 5 \text{ и.т.н.}$$

Пр. Биномни коефициенти

$$B_{n,k} = B_{n-1,k} + B_{n-1,k-1} \quad B_{n,0}=1, B_{n,n}=1, n \geq k > 0$$

Пр. НЗД

$$\text{НЗД}_{n,k} = n \quad \text{за } n=k$$

$$\text{НЗД}_{n,k} = \text{НЗД}_{n-k,k} \quad \text{за } n > k$$

$$\text{НЗД}_{n,k} = \text{НЗД}_{n,k-1} \quad \text{за } n < k$$

## 1. Рекурзивни функции

При пишувањето на функциите, главната функција ги повикува корисничките функции. Но една функција може да се повикува самата себе. Ова повикување се вика **рекурентно (повторно) повикување**. Таквата постапка е наречена **рекурзија**, а таквите функции се **рекурзивни функции**.

**Рекурзија** е техника на дефинирање или опишување на објект или процес во кој дефиницијата или описот се однесуваат на самиот објект или процес.

Рекурзивните алгоритми се потешко разбирливи и побавни од итеративните.

Пр. збир на првите  $n$  природни броеви

функција сума( $n$ );

почеток

ако  $n=0$

тогаш

врати  $\leftarrow 0$

инаку

врати  $\leftarrow$  сума( $n-1$ )+ $n$ ;

крај\_ако;

Пр. програма која со рекурзивна функција пресметува збир на природните броеви од 1 до  $n$ .

```
#include <iostream>
```

```
using namespace std;
```

```
int zbir(int n)
```

```
{
```

```
    if(n==0) return 0;
```

```
    else return zbir(n-1)+n;
```

```

}

int main()
{
    int n;
    cout << "vnesi go n" << endl;
    cin>>n;
    cout<<"zbirot e "<<zbir(n);
    return 0;
}

```

Пр. програма која со рекурзивна функција пресметува степен на природен број

```

#include <iostream>
using namespace std;

int stepen(int a, int n)
{
    if (n==1)
    {
        return a;
    }
    else return a * stepen(a, n-1);
}

```

```

int main()
{
    int n,a;
    cout<<"Vnesi osnova"<<endl;
    cin>>a;
    cout<<"Vnesi stepen"<<endl;
    cin>>n;
    cout<<"a na stepen n e "<<stepen (a,n);
    return 0;
}

```

Пр. програма која со рекурзивна функција пресметува факториел на природен број

```

#include <iostream>
using namespace std;

int faktoriel(int n)
{
    if (n==1)
    {
        return 1;
    }
    else return n * faktoriel(n-1);
}

```

```

int main()
{
    int n;
    cout<<"Vnesi broj"<<endl;
    cin>>n;
}

```

```

    cout<<"Faktoriel e " << faktoriel(n);
    return 0;
}

```

Пр. програма која со рекурзивна функција ги печати фибоначиевите броеви од 1 до n

```

#include <iostream>

using namespace std;
int fibonaci_niza(int broj)
{
    if(broj==0)
    {
        return 1;
    }
    if (broj==1)
    {
        return 1;
    }

    return fibonaci_niza(broj-1) + fibonaci_niza(broj-2);
}
int main()
{
    int n,i;
    cout << "vnesi go n" << endl;
    cin>>n;
    for(i=0;i<=n;i++)
        cout<<fibonaci_niza(i)<<" ";
    return 0;
}

```

Пр. програма која со рекурзивна функција пресметува збир на цифрите на природен број n.

```

#include <iostream>
using namespace std;

int zbir(int n)
{
    if (n<10) return n;
    else return n%10+zbir(n/10);
}

int main()
{
    int n;
    cout << "vnesi go n" << endl;
    cin>>n;
    cout<<"zbirot na cifrite e " << zbir(n);
    return 0;
}

```

Пр. програма која со рекурзивна функција пресметува дали производите на цифрите на два цели броја се исти. Производ на цифри да се одреди со рекурзија.

```

#include <iostream>
using namespace std;

int proizvod(int n)
{
    if (n<10) return n;
    else return n%10*proizvod(n/10);
}

int main()
{
    int n,m,a1,a2;
    cout << "vnesi go prviot broj" << endl;
    cin>>n;
    cout << "vnesi go vtoriot broj" << endl;
    cin>>m;
    a1=proizvod(n);
    a2=proizvod(m);
    if(a1==a2) cout<<"da";
    else cout<<"ne";
    return 0;
}

```

Пр. програма која со рекурзивна функција пресметува збир на цифрите на природен број  $n$ . Да се провери и испечати дали збирот е прост број(проверката дали бројот е прост направи ја со функција).

```

#include <iostream>
using namespace std;

int zbir(int n)
{
    if (n<10) return n;
    else return n%10+zbir(n/10);
}

void prost(int m)
{
    int i, t=0;
    for(i=2;i<=m/2;i++)
    {
        if (m%i==0)
        {
            t=1;
            break;
        }
    }
    if (t==0) cout<<"brojot e prost";
    else cout<<"brojot ne e prost";
}

int main()
{
    int n,k;
    cout << "vnesi go n" << endl;
    cin>>n;
}

```

```

k=zbir(n);
cout<<"zbirot na cifrite e "<<zbir(n)<<endl;
prost(k);
return 0;
}

```

Пр. програма која со рекурзивна функција пресметува дали збирот на цифрите на два цели броја се исти. Збир на цифри да се одреди со рекурзија.

Пр. програма која со рекурзивна функција пресметува збир на парните цифри на природен број  $n$ .

```

#include <iostream>
using namespace std;
int zbir(int n)
{
    int k;
    if (n==0) return 0;
    else
    {
        k=n%10;
        if(k%2==0)
            return k+zbir(n/10);
        else return 0+zbir(n/10);
    }
}

int main()
{
    int n;
    cout << "vnesi go n" << endl;
    cin>>n;
    cout << "zbirot na parnite cifri e " <<zbir(n)<< endl;

    return 0;
}

```

Пр. програма која со рекурзивна функција пресметува збир на непарните цифри на природен број  $n$ .

```

#include <iostream>
using namespace std;
int zbir(int n)
{
    int k;
    if (n==0) return 0;
    else
    {
        k=n%10;
        if(k%2!=0)
            return k+zbir(n/10);
        else return 0+zbir(n/10);
    }
}

int main()
{

```

```

int n;
cout << "vnesi go n" << endl;
cin>>n;
cout << "zbirot na neparnite cifri e " <<zbir(n)<< endl;

return 0;
}

```

Пр. програма која за внесен цел број ги печати само цифрите од бројот кои се помали од 5 и бројот на отпечатени цифри. Печатењето на цифрите на бројот помали од 5 да се реализира со рекурзивна функција.

```

#include <iostream>
using namespace std;
void rekurzija(int m, int b)
{
    int c;
    if(m==0)
    {
        cout<<endl;
        cout<<"ima " <<b<<" cifri";
    }
    else
    {
        c=m%10;
        if(c<5)
        {
            cout<<c<<" ";
            b++;
        }
        return rekurzija (m/10,b);
    }
    else
        return rekurzija (m/10,b);
}
}

```

```

int main()
{
    int m,i,b=0;
    cout << "vnesi go m" << endl;
    cin>>m;
    rekurzija(m,b);
    return 0;
}

```

Пр. програма која за n внесени цели броеви ги печати само цифрите од бројот кои се помали од 5 и бројот на отпечатени цифри за секој број. Печатењето на цифрите на бројот помали од 5 да се реализира со рекурзивна функција.

```

#include <iostream>
using namespace std;
int mali(int a)
{
    if(a==0)
    {

```

```

    return 0;
}
int cifra=a%10;
if(cifra<5)
{
    int b= 1 + mali(a/10);
    cout<<cifra<<" ";
    return b;
}
return 0 + mali(a/10);
}

int main()
{
    int i,n;
    cout<<"Vnesi go n"<<endl;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        int broj;
        cin>>broj;
        cout<<"brojot "<<broj<<" ima "<<mali(broj)<<" cifri pomali od 5"<<endl;
        cout<<endl;
    }
    return 0;
}

```

Пр. програма која со рекурзивна функција пресметува НЗД на два два природни броја.

```

#include <iostream>
using namespace std;

void NZD(int &a, int &b)
{
    int ost;
    ost=a%b;
    if(ost!=0)
    {
        a=b;
        b=ost;
        NZD(a,b);
    }
}

void zamena(int &a, int &b)
{
    int pom=a;
    a=b;
    b=pom;
}

int main()
{
    int m,n;
    cout << "vnesi dva celi brja " << endl;
    cin>>m>>n;
}

```

```

    if(m<n)
        zamena(m,n);
    NZD(m,n);
    cout<<"NZD e " <<m;
    return 0;
}

```

Пр. програма која со рекурзија го наоѓа индексот на првиот член на низа а со n елементи кој има вредност x. Ако таков елемент нема да се испечати вредноста -1.

```
#include <iostream>
```

```

using namespace std;
int baraj(int a[], int n, int x, int i)
{
    if(i>=n-1) return -1;
    if(a[i]==x) return i;
    return baraj(a,n,x,i+1);
}

```

```

int main()
{
    int n,i,a[100],x;
    cout << "vnesi go n" << endl;
    cin>>n;
    cout<<"vnesi gi elementite na nizata"<<endl;
    for(i=0;i<n;i++)
        cin>>a[i];
    cout << "vnesi go x" << endl;
    cin>>x;
    cout<<"pozicijata e " << baraj(a,n,x,0);
    return 0;
}

```

Пр. Да се напише програма која со рекурзивна функција со која се одредува најмал елемент на низа со цели броеви.

```

using namespace std;
int baraj(int a[], int n, int m, int i)
{
    if (i>=n) return m;
    else
    {
        if(a[i]<m) m=a[i];
        return baraj(a,n,m,i+1);
    }
}

```

```

int main()
{
    int n,i,a[100],m;
    cout << "vnesi go n" << endl;
    cin>>n;
    cout<<"vnesi gi elementite na nizata"<<endl;
}

```



```
for(i=0;i<n;i++)
    cin>>a[i];
m=a[0];
cout<<"najmal element e "<< baraj(a,n,m,0);
return 0;
}
```