

5

Рекурзивни функции

1. Да се напише програма која за дадено n ќе го испише соодветниот Фиbonачиев број.
Фиbonачиевите броеви се дефинирани на следниов начин:

$$\begin{aligned}a_1 &= 1 \\a_2 &= 1 \\a_n &= a_{n-1} + a_{n-2}\end{aligned}$$

Пример: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

```
#include <stdio.h>

int fib(int n);
int main()
{
    int broj;
    printf("Vnesi broj:\n");
    scanf("%d", &broj);
    printf("Fibonacieviot broj e: %d\n", fib(broj));
    return 0;
}
int fib(int n)
{
    if (n == 1 || n == 2) return 1;
    else return fib(n - 1) + fib(n - 2);
}
```

2. Да се напише програма која содржи функција за пресметување на факториел од даден број:

```
#include <stdio.h>
int factorial(int n);
int main ()
{
    int broj;
    printf("Vnesi broj:\n");
    scanf("%d", &broj);
    printf("%d!=%d\n", broj, factorial(broj));
    return 0;
}
int factorial(int n)
{
    if (n==0)
        return 1;
    else
        return n*factorial(n-1);
}
```

3. Да се напише рекурзивна функција која ќе пресметува најголем заеднички делител на два дадени броја.

```
#include <stdio.h>

int nzd(int i,int j);
int main()
{
    int n,m;
    float x0=1.0, x1=2.0;
    printf("Vnesi dva broja:\n");
    scanf("%d %d", &n, &m);
    printf("Najgolemiot zaednicki delitel na ");
    printf("broevite %d i %d e %d.\n",n,m,nzd(n,m));
    return 0;
}
int nzd(int i, int j)
{
    int vrednost;
    if(i!=j)
        if(i<j)
            vrednost=nzd(j-i,i);
        else
            vrednost=nzd(i-j,j);
    else
        vrednost = i;
    return vrednost;
}
```

4. Да се напише програма што ќе ја испишува вредноста на произволен член на низата дефинирана со:

$$\begin{aligned}x_1 &= 1 \\x_2 &= 2 \\x_n &= \frac{n-1}{n} x_{n-1} + \frac{1}{n} x_{n-2}\end{aligned}$$

```
#include <stdio.h>
float xnn(float x1,float x2,int k);
int main(){
    int n;
    float x1=1.0, x2=2.0;
    printf("Vnesi broj:\n");
    scanf("%d", &n);
    printf("%d-tiot clen na nizata e: %f.\n",n,xnn(x1,x2,n));
    return 0;
}
float xnn(float x1,float x2,int k) {
    float clen;
    if(k==1)
        clen=x1;
    else
        if(k==2)
            clen=x2;
        else
            clen=xnn(x1,x2,k-1) * (k-1) / k+xnn(x1,x2,k-2) * 1 / k;
    return clen;
}
```

5. Да се напише рекурзивна функција која од даден цел број ќе врати колку цифри има бројот.

```
#include <stdio.h>

int br_cifri(int x) {
    if(x==0) return 0;
    return 1+br_cifri(x/10);

}
```

Функцијата нема да работи за одредени вредности. За кои броеви нема да работи правилно функцијата?

6. Да се напише рекурзивна функција која од даден цел број ќе врати сумата на цифрите на тој број.

```
#include <stdio.h>

int suma_cifri(int x) {
    if(x==0) return 0;
    return x%10+suma_cifri(x/10);

}
```

Втор начин: со користење на дополнителна променлива во која ќе се чува сумата на цифрите. Променливата мора да биде иницијализирана на нула во првиот повик на функцијата.

```
#include <stdio.h>
int suma_cifri(int x,int suma) {
    if(x==0) return suma;
    return suma_cifri(x/10,suma+x%10);
}
int main()
{
    int broj;
    printf("Vnesete broj\n");
    scanf("%d",&broj);
    printf("Zbirot na cifrite na brojot %d e %d\n",broj,suma_cifri(broj,0));
    return 0;
}
```

7. Да се напише рекурзивна функција која од даден цел број ќе ја врати сумата од ASCII кодовите на цифрите на даден број.

```
#include <stdio.h>

int suma_ascii(int x) {
    if(x==0) return 0;
    return suma_ascii(x/10)+x%10+'0';

}
```

Бидејќи ASCII табелата чува различна вредност за 1 како цел број и '1' како знак и исто така ASCII кодовите се последователни кај цифрите, затоа $2+'0'='2'$ т.е. $x+'0'='x'$ под услов x да е цифра.

Доколку функциите од задача 5 и 6 се повикаат со истиот број, резултатот ќе е следниот:

```
suma_cifri(1234) == 10
suma_ascii(1234) == 202
```

8. Да се напише рекурзивна функција која од даден број ќе ја врати цифрата со најголема вредност.

Претпоставуваме дека последната (најлевата) цифра е најголемата, па потоа се обидуваме да најдеме некоја која што е поголема од таа.

```
int max_cif(int x){
    int m,cif;
    if(x/10==0)
        return x; // ja vrakjame najlevata cifra
    cif=x%10; // tekovna cifra
    m=max_cif(x/10); // najgolema vo ostatokot od brojot
    if(m>cif)
        return m;
    else
        return cif;
}
```

Втор начин: максималната цифра ќе ја пренесеме како втор аргумент (на почеток тоа ќе биде последната цифра од бројот), а потоа во рекурзивната функција со секоја најдена цифра поголема од максималната, вредноста на вториот аргумент ќе ја менуваме.

```
#include <stdio.h>
int max(int broj, int maxCif)
{
    if(broj==0) return maxCif;

    if(broj%10>maxCif)
        max(broj/10,broj%10);
    else
        max(broj/10,maxCif);
}
int main()
{
    int broj;
    printf("Vnesete broj\n");
    scanf("%d",&broj);
    printf("Maksimalnata cifra vo brojot %d e %d\n",broj,max(broj/10,broj%10));
    return 0;
}
```

9. Да се напише рекурзивна функција која за даден цел број ќе провери дали има барем една цифра што е делива со 3.

```
int proverka(int x){
    if(x==0) return 0;
    if((x%10)%3==0)
        return 1;
    return proverka(x/10);
}
```