

## Упражнение №9

### ФУНКЦИИ С ПАРАМЕТРИ, ПРЕДАВАНИ ПО СТОЙНОСТ

Функцията е самостоятелна програмна единица, при изпълнението на която се получават определени резултати.

Функцията се дефинира по следния начин:

```
<тип_на_резултат> <идентификатор> (<списък от формални параметри>)  
{  
    <описание на локални променливи>;  
    <оператор>  
    . . .  
    return <израз>;  
}
```

Функцията се декларира само със заглавния си ред (прототип на функцията):

```
<тип_на_резултат> <идентификатор> (<списък от формални параметри>);
```

Функцията се извиква с името си, следвано от списък от фактически параметри, разделени със запетая и затворени в кръгли скоби; броят, типът и последователността на фактическите параметри трябва точно да отговарят на броя, типа и последователността на формалните такива:

```
<идентификатор> (<списък от фактически параметри>);
```

Ако функцията връща резултат, тя трябва да бъде извикана или вдясно от операция присвояване, или в израз, или във функция **printf()**. Ако функцията не връща резултат, тя се извиква на отделен ред.

Пример 1. Да се състави функция, която не връща резултат, а само изчислява и отпечатва сумата от четните цели числа, разположени в интервала [a, b]; границите на интервала се подават като параметри на функцията. (Тъй като не връща резултат, функцията се извиква на отделен ред.)

```
#include <stdio.h>  
void Sum(int a, int b); //декларация на функцията  
int main()  
{  
    int low, high;  
    printf("Vavedete dolna granica: ");  
    scanf("%d", &low);  
    printf("Vavedete gornagranica: ");  
    scanf("%d", &high);  
    Sum(low, high); //извикване на функцията на отделен ред  
    return 0;  
}  
void Sum(int a, int b) //дефиниция на функцията  
{  
    int s = 0, num;  
    for(num = a; num <= b; num++)  
    {  
        if((num % 2) == 0)  
            s += num;  
    }  
    printf("Suma ot chetni chisla v intervala [%d-%d]=%d\n", a, b, s);  
}
```

Пример 2. Да се състави функция, която няма параметри, въвежда от клавиатурата цяло число, намира и връща като резултат броя на цифрите на числото. (Функцията се извиква в цикъл, докато потребителят иска това.)

```
#include <stdio.h>
int Cifri();
int main()
{
    int otgovor;
    do
    {
        //функцията се извиква многократно - в цикъл
        //освен това функцията се извиква от функция printf()
        printf("Vavedenoto chislo ima %d cifri\n", Cifri());
        printf("Iskate li da prodaljite (1 = da): ");
        scanf("%d", &otgovor);
    }
    while(otgovor == 1);
    return 0;
}
int Cifri()
{
    int chislo, broj;
    broj = 0;
    printf("Vavedete cyalo chislo: ");
    scanf("%d", &chislo);
    do
    {
        chislo = chislo / 10;
        broj++;
    }
    while(chislo != 0);
    return broj;
}
```

Пример 3. Да се състави функция, която получава като параметри: (x,y)-координати на центъра на окръжност, радиус на окръжността и (x,y)-координати на точка в равнината. Функцията да връща резултат 0 – ако точката лежи върху окръжността, резултат +1 – ако точката лежи извън окръжността и резултат -1 – ако точката лежи вътре в окръжността. (Примерът илюстрира извикване на функция в израз на оператор **switch**.)

```
#include <stdio.h>
#include <math.h>
int PointToCircle(float xc, float yc, float r, float x, float y);
int main()
{
    float circleX, circleY, radius, x, y;
    printf("Vavedete x-coordinatata na centara na okrajnostta: ");
    scanf("%f", &circleX);
    printf("Vavedete y-coordinatata na centara na okrajnostta: ");
    scanf("%f", &circleY);
    printf("Vavedete radius: ");
    scanf("%f", &radius);
    printf("Vavedete x-coordinatata na tochkata: ");
    scanf("%f", &x);
    printf("Vavedete y-coordinatata na tochkata: ");
    scanf("%f", &y);
}
```

```

        //функцията се извиква в израз на оператор switch
        switch(PointToCircle(circleX, circleY, radius, x, y))
        {
            case 0: printf("Tochkata leji varhu okrajnostta\n"); break;
            case 1: printf("Tochkata leji izvan okrajnostta\n");break;
            case -1: printf("Tochkata leji vatre v okrajnostta\n");break;
        }
        return 0;
    }
}

int PointToCircle(float xc, float yc, float r, float x, float y)
{
    double dist;
    dist = sqrt(pow((x-xc), 2)+pow((y-yc),2));
    printf("Dist = %lf\n", dist);
    if(dist == r)
        return 0;
    else
        if(dist > r)
            return 1;
        else
            return -1;
}

```

**Пример 4.** Да се състави функция, която при зададени като параметри стойности на променливите **x**, **h** и **n** изчислява следната числова редица:

$$S = (x+h)/3 + (x+2h)/5 + (x+3h)/7 + \dots + (x+n.h)/(2.n+1)$$

```

#include <stdio.h>
float Redica(float x, float h, int n);
int main()
{
    float x, h;
    int broj;
    printf("Vavedete x: ");
    scanf("%f", &x);
    printf("Vavedete h: ");
    scanf("%f", &h);
    printf("Vavedete broj chlenove na redicata: ");
    scanf("%d", &broj);
    printf("Stojnost na redicata = %f\n", Redica(x, h, broj));
    // ili
    // float rezultat = Redica(x, h, broj);
    // printf("Stojnost na redicata = %f\n", rezultat);
    return 0;
}

float Redica(float x, float h, int n)
{
    float sum, chislitel;
    int i, znamenatel;
    sum = 0;
    i = 1;
    chislitel = x;
    znamenatel = 1;
    while(i <= n)
    {
        chislitel += h;
        znamenatel += 2;
        sum = sum + chislitel/znamenatel;
        i++;
    }
    return sum;
}

```