

Функции в C++

Досега използвахме функцията `main()`, а също и някои други готови функции, например `swap(a,b)`

При почти всички програми се оказва, че е много удобно да се използват и други функции, създадени от програмистите.

Причината е, че често се налага една и съща последователност да се използва повече от веднъж.

Функция без параметри

Пример: Няколкократно пресмятане на периметър и лице на правоъгълник със страни a и b.

//пресмятане за един правоъгълник

```
int L = 2*(a+b); cout << L << endl;
```

```
int S = a*b; cout << S << endl;
```

Оформяне като функция

```
void p()  
{  
    int L = 2*(a+b); cout << L << endl;  
    int S = a*b; cout << S << endl;  
}
```

Използване на функцията

```
int a,b;
```

```
cin >> a >> b;
```

```
p();
```

```
// промяна на размерите, например удвояване
```

```
a=2*a; b=2*b;
```

```
p();
```

Функция с параметри

```
void p(int x, int y)
{
    int L = 2*(x+y); cout << L << endl;
    int S = x*y; cout << S << endl;
}
```

Използване:

```
int a,b; cin >> a >> b;
p(a,b);
```

Функция, която връща стойност

```
int f(int a, int b)
{
    int c;
    if(a>b) c = a; else c=b;
    return c;
}
```

Използване:

```
int a,b; cin >> a >> b;
int r=f(a,b); cout << r;
или
cout << f(a,b);
```

Общ вид на дефиниция на функция

<стойност, която връща> <име на функцията>
(<списък от параметри>)
{<тяло, т.е. списък от оператори>}

Примери за дефиниции на функции:

```
void f1(){cout << x;}
```

```
int f2(){return x*x;}
```

```
void f3(){int a; cin >> a; cout << a*a;}
```

```
int f4(int a) { int b=a*a; return b;}
```

```
int f5(int a,int b) { int c=a*b*x; cout << c; return c;}
```

Примери за използване на функции:

```
int x=5;
```

```
void f1(){cout << x;}
```

```
int main()
```

```
{ f1();}
```



```
int x;
```

```
int f2(){return x*x;}
```

```
int main()
```

```
{ x=5;
```

```
  cout << f2();
```

```
}
```

```
void f3(){int a; cin >> a; cout << a*a;}
```

```
int main()
```

```
{
```

```
    f3();
```

```
}
```

```
int f4(int a) { int b=a*a; return b;}
```

```
int main()  
{  
    int a  
    cin >> a;  
    f4(a);  
}
```

```
int x;
```

```
int f5(int a,int b)
```

```
{int c=a*b*x; cout << c << endl; return c;}
```

```
int main()
```

```
{
```

```
    int y,z;
```

```
    cin >> x >> y >> z;
```

```
    cout << 6*f5(y,z)+7 << endl;
```

```
}
```

Аналогия с функция с математика

Когато функцията в C++

- не използва и не променя глобални променливи
- няма странични ефекти (напр. няма cout)
- връща стойност (не е void)

Пример:

```
int f(int x,int y){return x*x+y*y;}
```

```
int main(){int r=5*f(6,7)+8; cout << r;}
```

Видове променливи

- Глобални
- Локални
- Параметри

Приоритети при употреба на еднакви имена

Пример:

```
int a=5;
```

```
void f1(){int a=6; cout << a << endl;}
```

```
и извикване: f1(); cout << a << endl;
```

ще отпечата

6

5

Пример:

```
int a=5;
```

```
void f2(){a=6; cout << a << endl;}
```

и извикване `f2(); cout << a << endl;`

ще отпечата

6

6

Пример:

```
int a=5;
```

```
void f3(int a){a=6; cout << a << endl;}
```

```
и извикване f3(a); cout << a << endl;
```

ще отпечата:

6

5

Пример за грешка при компиляция:

```
int a=5;
```

```
void f4(int a){int a=6; cout << a << endl;}
```

Запознаване с алгоритмите (начало за теория на числа)

Отделяне на цифрите на цяло положително число:

```
void f(int a)
{
    do
    {
        cout << a%10 << endl;
        a = a/10;
    }
    while(a>0);
}
```

Съставяне на число от неговите цифри

```
int f()
{ int n; cin >> n;
  int a=0;
  for(int i=1;i<=n;i++)
  { int b; cin >> b;
    a = 10*a+b;  // схема на Хорнер
  }
  return a;
}
int main() { cout << f() << endl;}
```

Развитие на $1/n$ във вид на десетична дроб с дължина k десетични разряда

```
void d(int n, int k)
{
    int r=10;
    for(int i=1; i<=k; i++)
    {
        int c=r/n;
        cout << c;
        r = r%n;
        r = 10*r;
    }
}

int main()
{int n,k; cin >> n >> k; d(n,k);}
```

Делимост на числата

Положителни делители на цяло положително число

```
int a; cin >> a;
```

```
for(int i=1; i<=a; i++)
```

```
if(a%i==0) cout << i << endl;
```

Просто число

Разлагане на прости множители на числото N

```
int N; cin >> N;
int p=2;
while(p<=N)
{
    while(N%p==0)
    {
        cout << p << " "; N = N/p;
    }
    p++;
}
```

Проверка, дали едно число е просто

```
bool p(int a) // a>1
{
    for(int i=2;i<a;i++)        // или for(int i=2;i*i<=a;i++)
        if(a%i==0){return false;}
    return true;
}
```

```
int main()
{
    int a; cin >> a;
    cout << p(a) << endl; // отпечатва 0 или 1
}
```

Най-голям общ делител на a и b

//наивен алгоритъм

```
int nod(int a, int b)
{
    int d,i=1;
    while((i<=a)&&(i<=b))
        { if((a%i==0)&&(b%i==0)) d=i; i++;}
    return d;
}
```


Най-голям общ делител на a и b

```
//алгоритъм на Евклид  
int nod_euclid(int a, int b)  
{  
    while( a!=b)  
        if(a>b)a=a-b;else b=b-a;  
    return a;  
}
```