



实验舱  
青少年编程  
走近科学 走进名校

# 挑战信息学奥林匹克

## C++程序设计 (6) 函数

# 认识函数

```
int main()  
{  
    return 0;  
}
```

```
int abs(int i)  
double fabs(double x)  
double ceil(double x)  
double sqrt(double x)
```

## 模块化程序设计

一个复杂问题，肯定是由若干稍简单的问题构成。模块化是把程序要解决的总目标分解为子目标，再进一步分解为具体的小目标，把每一个小目标称为一个模块。

---

# 函数的定义

数据类型 函数名(形式参数列表)

{

函数体;

}

```
int Abs(int x)
{
    if (x >= 0) return x;
    else return -x;
}
```

## 说明

1. 函数名的命名规则按照变量的命名规则命名。
2. 如果函数有返回值，必须用**return**返回，返回值由返回值类型规定其返回类型。

# 函数的定义

数据类型 函数名(形式参数列表)

{  
    函数体;  
}

```
void Swap(int &x, int &y)
{
    int t;
    t=x; x=y; y=t;
}
```

说明

3. 如果函数没有返回值，在“返回值类型用void，函数体内可以不用return返回值。
4. 形式参数可以多个，也可以没有。

# 有值函数调用

自定义函数放在main函数之前，在main函数中可以直接调用。

有值函数只能在表达式中调用

```
#include <bits/stdc++.h>
using namespace std;
int Abs(int x)
{
    if (x >= 0) return x;
    else return -x;
}
int main()
{
    int n;
    cin >> n;
    cout << Abs(n) << endl;
    return 0;
}
```

# 无值函数调用

无值函数单独调用

```
#include <bits/stdc++.h>
using namespace std;
void Swap(int &x, int &y)
{
    int t;
    t=x; x=y; y=t;
}
int main()
{
    int n, m;
    cin >> n >> m;
    Swap(n, m);
    cout << n << " " << m << endl;
    return 0;
}
```

# 函数申明

自定义函数放在main函数之后，在调用之前申明。

```
#include <bits/stdc++.h>
using namespace std;
void Swap(int &x, int &y);
int main()
{
    int n, m;
    cin >> n >> m;
    Swap(n, m);
    cout << n << " " << m << endl;
    return 0;
}
void Swap(int &x, int &y)
{
    int t;
    t=x; x=y; y=t;
}
```

---

# 函数传值

形参，传值

```
#include <bits/stdc++.h>
using namespace std;
int Abs(int x)
{
    if (x >= 0) return x;
    else return -x;
}
int main()
{
    int n;
    cin >> n;
    cout << Abs(n) << endl;
    return 0;
}
```

实参

形参，传址

```
#include <bits/stdc++.h>
using namespace std;
void Swap(int &x, int &y)
{
    int t;
    t=x; x=y; y=t;
}
int main()
{
    int n, m;
    cin >> n >> m;
    Swap(n, m);
    cout << n << " " << m << endl;
    return 0;
}
```

实参



# 局部变量

```
void Swap(int x, int y)
{
    int t;
    t = x; x = y; y = t;
}
int main()
{
    int x, y;
    cin >> x >> y;
    Swap(x, y);
    cout << x << " " << y << endl;
    return 0;
}
```

定义在函数内部的变量称为局部变量。局部变量的作用域是从变量定义的位置一直到函数结束的范围。

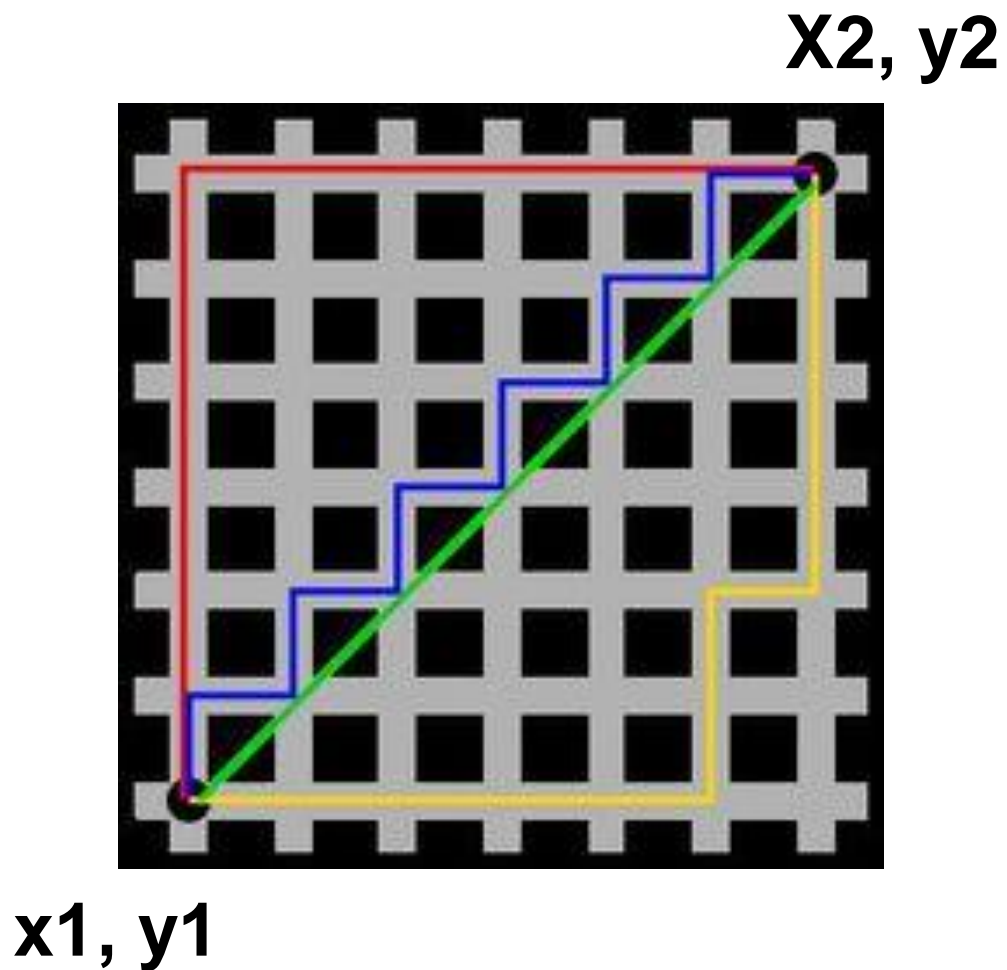
# 全局变量

```
int x, y;
void Swap()
{
    int t;
    t = x; x = y; y = t;
}
int main()
{
    cin >> x >> y;
    Swap();
    cout << x << " " << y << endl;
    return 0;
}
```

在函数以外并在函数之前定义的变量称为全局变量。全局变量的作用域是从变量定义的位置一直到程序结束的范围。

本例中x、y的作用范围是整个程序，虽然没有参数传递，Swap函数可以直接对x、y操作。

例题-1：给出平面上两个点的坐标，求两点的曼哈顿距离。



■ 曼哈顿距离

$$|x_1 - x_2| + |y_1 - y_2|$$

$$\text{Abs}(x_1 - x_2) + \text{Abs}(y_1 - y_2)$$

## 参考代码

```
double Abs(double x)
{
    if ( x >= 0 ) return x;
    else return -x;
}
int main()
{
    double x1, x2, y1, y2, x;
    cin >> x1 >> y1 >> x2 >> y2;
    x = Abs(x1-x2) + Abs(y1-y2);
    printf("%.3lf\n", x);
    return 0;
}
```

---

## 例题-2：石头剪子布

现在有两个人在玩石头剪子布游戏，请你判断最后谁赢了。

用  $R$  代表石头， $S$  代表剪子， $P$  代表布。

### 输入

输入的第一行是一个整数  $t$  ( $0 < t < 1000$ )，表示测试样例的数目。

每组输入样例的第一行是一个整数  $n$  ( $0 < n < 100$ )，表示游戏次数。

接下来  $n$  行，每行由两个字母组成，两个字母之间用一个空格分隔，这些字母只会是 `R`，`S` 或 `P`。第一个字母表示 *Player1* 择，第二个字母表示 *Player2* 的选择。

### 输出

对于每组输入样例，输出获胜方的名字 (*Player1* 或 *Player2*)，如果平局，则输出 *TIE*。

### 样例输出

Player 2

TIE

Player 1

### 样例输入

3

2

R P

S R

3

P P

R S

S R

1

P R

# 算法分析

R – 石头, S – 剪子, P – 布

player1 player2 结果

R S 1

S R 2

S R 1

P S 2

R P 2

P R 1

平局的结果用0表示

设计一个函数, 判断输赢

```
int Iswin(char p1, char p2)
{
    if (p1 == 'R' && p2 == 'S') return 1;
    if (p1 == 'S' && p2 == 'R') return 2;
    if (p1 == 'R' && p2 == 'P') return 2;
    if (p1 == 'P' && p2 == 'R') return 1;
    if (p1 == 'S' && p2 == 'P') return 1;
    if (p1 == 'P' && p2 == 'S') return 2;
    if (p1 == p2) return 0;
}
```

# 算法分析

- 统计每局中双方输赢的次数

```
for (int j = 0; j < n; j++)  
{  
    cin >> c1 >> c2;  
    if (Iswin(c1, c2) == 1) s1++;  
    else if (Iswin(c1, c2) == 2) s2++;  
}
```

---

# 算法分析

- 根据统计结果，确定输赢的结果

```
if (s1 == s2)
    cout << "TIE" << endl;
else if (s1 > s2)
    cout << "Player 1" << endl;
else
    cout << "Player 2" << endl;
```

---



## 参考程序

```
for (int i = 0; i < t; i++)
{
    cin >> n;
    s1 = 0; s2 = 0;
    for (int j = 0; j < n; j++)
    {
        cin >> c1 >> c2;
        if (Iswin(c1, c2) == 1) s1++;
        else if (Iswin(c1, c2) == 2) s2++;
    }
    if (s1 == s2) cout << "TIE" << endl;
    else if (s1 > s2) cout << "Player 1" << endl;
    else cout << "Player 2" << endl;
}
```

---

## 例题-3: 回文数素数

### 【问题描述】

如果一个数从左边读和从右边读都是同一个数，就称为回文数。例如6886就是一个回文数，从给出的数据中统计出既是回文数又是素数的数。

### 【输入】

输入一个整数 $n$ ，下面一行有 $n$ 个正整数。

### 【输出】

输出为一个数，统计出的个数。

### 【输入样例】

7

7 12 10 11 121 1331 10301

### 【输出样例】

3

---

## 判断质数的函数

```
bool Isprime(long long n)
{
    if (n < 2) return false;
    long long m = sqrt(n);
    for (long long i = 2; i <= m; i++)
    {
        if (n % i == 0) return false;
    }
    return true;
}
```

---

## 判断回文数函数

```
bool Ishuiwen(long long n)
{
    long long t, m = 0;
    t = n;
    while (t > 0)
    {
        m = m * 10 + t % 10;
        t /= 10;
    }
    if (m == n) return true;
    else return false;
}
```

---

# 算法分析

- 循环读取数据x
- 判断x是否回文和质数，并统计

```
for (int i = 0; i < n; i++)  
{  
    cin >> x;  
    if (Ishuiwen(x) && Isprime(x))  
    {  
        ans++;  
    }  
}
```

## 例题-4：n个数的最小公倍数

### 【题目描述】

两个或多个整数公有的倍数叫做它们的公倍数，其中除0以外最小的一个公倍数就叫做这几个整数的最小公倍数。

### 【输入描述】

第一行一个数 $n$  ( $1 \leq n \leq 10$ )

接下来 $n$ 行，每行一个整数，（保证每个数小于  $10^9$ ）

### 【输出描述】

输出这 $n$ 个数的最小公倍数

### 【输入样例】

2

18 12

### 【输出样例】

36

---

# 算法分析

- 2个数的公倍数

$$G = x * y / \gcd(x, y)$$

- 3个数的公倍数

先求出其中两个数的最小公倍数，再求这个最小公倍数与第三个数的最小公倍数。

- n个数的公倍数

$x = G;$

$\text{cin} \gg y;$

$G = x * y / \gcd(x, y);$

---

# 最大公约数函数

```
int Gcd(int x, int y)
{
    int r = x % y;
    while ( r != 0 )
    {
        x = y;
        y = r;
        r = x % y;
    }
    return y;
}
```

---



## 参考程序

```
s = x;  
for ( int i = 1; i < n; i++ )  
{  
    cin >> y;  
    s *= y;  
    s /= Gcd(x, y);  
    x = s;  
}
```

---