

# 常用函数

## 数字转字符串(int->string)

```
1 string Change1(int x){  
2     if(x==0) return "0";  
3     string A;  
4     while(x){  
5         A+=((x%10)+'0');  
6         x/=10;  
7     }  
8     reverse(A.begin(),A.end());  
9     return A;  
10 }
```

注：其中  $x$  为非负整数，返回的  $A$  为字符串。

## 字符串转数字(string->int)

```
1 int Change2(string x){  
2     int cnt=0;  
3     for(int i=0;i<(int)x.size();i++){  
4         cnt*=10;  
5         cnt+=(x[i]-'0');  
6     }  
7     return cnt;  
8 }
```

注：其中  $x$  为非空字符串，返回的  $cnt$  为非负整数。

## 保留 $n$ 位小数输出

```
1 cout<<fixed<<setprecision(n)<<x;
```

注：其中  $x$  为 double 类型的浮点数，参数中  $n$  需填入一个非负整数。

## 读入一整行

```
1 getline(cin,s);
```

注：其中  $s$  需为一个 string 类型的变量。

## 判断质数

```
1 | bool prime(int a)
2 |
3 |     if(a<=1) return 0;
4 |     bool f=1;
5 |     for(int i=2;i*i<=a;i++) if(!(a%i)) f=0;
6 |     return f;
7 | }
```

注：其中  $a$  为非负整数。函数返回值为 1(true) 时，则  $a$  为质数，返回值为 0(false) 时，则  $a$  不为质数。

## 判断某个字符是什么

```
1 | int check(char x){
2 |     if(x>='0'&&x<='9') return 1;//x是数字
3 |     if(x>='A'&&x<='Z') return 2;//x是大写字母
4 |     if(x>='a'&&x<='z') return 3;//x是小写字母
5 |     return 0;//x既不是数字也不是字母
6 | }
```

## 固定占位输出

```
1 | cout<<setw(n)<<setfill(m)<<x;
```

注：`setw(n)` 表示该次输出固定占  $n$  位，`setfill(m)` 表示若输出量不够  $n$  位，不够的位用  $m$  字符填充(此处  $m$  为一个 `char` 类型变量)，`x` 表示要输出的内容。

例一：

```
1 | int n=5;char m='0';
2 | cout<<setw(n)<<setfill(m)<<123;
3 | /*
4 |  输出:
5 | 00123
6 | */
```

例二：

```
1 | int n=4;char m=' ';
2 | cout<<setw(n)<<setfill(m)<<"ab";
3 | /*
4 |  输出:
5 |  ab
6 | */
```

## 常用数学函数

```
1 | sqrt(n); //取非负整数 n 的平方根
2 | n=min(a,b); //返回 a 与 b 的最小值
3 | n=max(a,b); //返回 a 与 b 的最大值
4 | swap(a,b); //交换 a 与 b 的值
5 | n=pow(a,b); //返回 a 的 b 次方
```

## 数组中的常用函数(以 int 数组为例)

```
1 | int a[100005];
2 | sort(a+1,a+1+n)//把a[1~n]内所有数从小到大排序
3 | sort(a,a+n)//把a[0~(n-1)]内所有数从小到大排序
4 | reverse(a+1,a+1+n)//翻转a[1~n]内所有数, 翻转后的a[1]为原先a[n]的值, a[2]为原先a[n-1]
   的值
```