

一些常见函数整理

函数setw()

功能

函数setw(int n)是c++中在输出操作中使用的字段宽度设置，设置输出的域宽，n表示字段宽度。**只对紧接着的输出有效**，紧接着的输出结束后又变回默认的域宽。

当后面紧跟着的输出字段长度小于n的时候，在该字段前面用空格补齐；当输出字段长度大于n时，全部整体输出。

头文件

头文件为

其中io代表输入输出，manip是manipulator（操纵器）的缩写

iomanip的作用：

主要是对cin,cout之类的一些操纵运算符，比如

setfill,setw,setbase,setprecision等等。它是I/O流控制头文件，就像C里面的格式化输出一样。

示例

```
#include <iostream>
#include <iomanip>

using namespace std;

int main()
{
    cout << setw(3) << 1 << 1234 << endl;
    cout << 1234 << setw(7) << 1234 << 12345 << endl;
    cout << 1234 << setw(3) << 12345 << endl;
    cout << 1234 << setw(4) << 123456 << 12 << setw(5) << 123 << endl;
    return 0;
}
```

运行结果如下

```
Microsoft Visual Studio 调试 × + ▾  
11234  
1234 123412345  
123412345  
123412345612 123
```

字符串处理函数

函数strlen()

原型

```
// 在Visual C++ 6.0中, 原型为  
size_t strlen(const char *string);
```

头文件: string.h

功能: 返回字符串s的长度, 不包括'\0'

示例

```
#include <iostream>  
using namespace std;  
#include <string.h>  
int main(void)  
{  
    char s[] = "Golden Global View";  
    cout << s << " has " << strlen(s) << " chars";  
    return 0;  
}  
// Golden Global View has 18 chars
```

与sizeof()的区别

strlen()求的是字符串的实际长度, 遇到第一个'\0'停止

sizeof()返回的是变量声明后所占的内存数, 不是实际长度, 此外sizeof只是一个操作符

函数strcpy()

这个打出来如果报错需要换成strcpy_s()

原型

```
char* strcpy(char* destination, const char* source);
```

头文件: string.h

功能: 'strcpy' 函数会将源字符串复制到目标字符串, 直到遇到源字符串的空字符 ('\0') 为止, 包括空字符在内。复制完成后, 目标字符串将成为源字符串的副本。

示例

```
#include <iostream>
using namespace std;
#include <string.h>
int main() {
    char source[] = "Hello, World!";
    char destination[20];

    strcpy_s(destination, source);
    cout << "Source: " << source << endl;
    cout << "Destination: " << destination << endl;

    return 0;
}
```

函数strcat()

这个打出来如果报错需要换成strcat_s()

原型

```
char *strcat(char *str1, const char *str2);
```

头文件: string.h

功能: 把str2所指字符串添加到str1结尾处 (覆盖str1结尾的'\0')

示例

```
#include <iostream>
using namespace std;
#include <string.h>
int main() {
    char str1[20] = "Hello, ";
    char str2[] = "world";
    strcat_s(str1, str2);
```

```
    cout << "str1:" << str1 << endl;  
    return 0;  
}
```

函数strcmp()

原型

```
int strcmp(const char *str1, const char *str2)
```

头文件: string.h

功能: 比较两个字符串, 返回比较结果, 如果str1与str2字符串相等, 函数值返回0, 如果不想等, 返回值为两字符串中第一个不想等字符的ASCII码值之差