

字符串处理

主要内容

- C 字符串——字符数组
 - 输入输出
 - ■常用字符串处理函数
- C++ 字符串——string 对象
 - string 简介
 - cin 和 cout
- 相关例题

C 字符串——字符数组

- 输入函数
 - scanf("%s", s) 读入字符串,在第一个空白符(''、'\n'、'\t')处停止,不读入空白符,在串尾自动添加'\0'。
 - gets(s) 读入一行字符,直到遇到'\n',读入换行符并将其舍弃,在串尾自动添加'\0'。
 - c = getchar()
 scanf("%c", &c)
 读入下一个字符, 包括空白符。



下面的代码中,变量a和c读入的是123和'a'吗?

```
int a;
char c;
scanf(''%d%c'', &a, &c);
```

```
Input
123
a
```



要"吃掉"第一行末尾的回车。

• 可以赋值给一个临时变量

• ''%*c''表示读入然后舍弃

Code

Code



■ 输入小结

- scanf("%d") 和 scanf("%s") 是 "空白符不敏感" 的,无论有多少空白符都会忽略。
- gets()和 getchar()能力更"强",空白符也会读入使用时需留心。
- 如果读入的字符串中没有空格,建议用scanf("%s"),若需要读入空格,则必须用gets()。
- 两者混用时要明确每一个空格和回车的去向。



- 输出函数
 - printf() 格式化输出
 - puts(s) 输出s串,并在最后自动添加回车'\n'

```
int x = 123456;
int y = 1234567890;

printf("x=: %010d\n", x);
printf("y=: %010d\n", y);

int x = 123456;
int y = 1234567890;

printf("x=: %010d\n", x);
printf("y=: %010d\n", y);
```



- 常用字符串处理函数
 - 头文件 #include <string.h>
 - strlen(s)

最常用的字符串函数,一般紧跟在输入之后,读完字符串后马就上测长度。

strcmp(s1, s2)

用来比较两个字符串字典序的大小,即从左到右依次比较每个字符的 ACSII 码值,一旦分出大小就不再比较之后的字符。与两个串的长度无关。



- 从内存层面快速初始化
 - memset()

初始化为全 0: memset(a, 0, sizeof(a));

初始化为全-1: memset(a, -1, sizeof(a));

初始化为无穷大: memset(a, 0x3f, sizeof(a));

■ 变量和数组定义后,如果不读入数据,就一定要初始化

C ++字符串——string对象

- C++ 中有专门用来存储和处理字符串的类 string, 它会根据实际字符串的长度动态分配内存 , 在使用上带来了极大的方便。
- 大家把"类"理解为是一种数据类型就好,而"对象"就是创建的具体变量。因此可以把一个string 字符串完全当作一个普通变量对待。

- 通过点运算符。获取字符串的属性,例如字符串 长度
 - int len = str.length;
- string 对象又具有字符数组的属性,可以通过下标运算符[]访问其中的每一个字符
 string str = "Hello";
 char c = str[2];
- 使用string类需包含头文件string #include <string> 并在代码开头添加 using namespace std;



■ string 示例

Code

```
string s1;//默认初始化为空串string s3 = ''abc''; //定义时初始化为指定字符串char s[10] = ''Hello'';string s4(s);//C字符串转C++字符串char c = s4[1];//取出单个字符s1 = s3 + ''123'';//+用于连接两个字符串s2 += s1;//+=将另一个串连接到末尾bool f = s2 < s3;</td>//<和>直接用于字符串比较
```



cin 输入和 cout 输出

- C++ 专门的输入输出方法是cin和cout,它兼容 C 的所有数据类型,包括字符数组。
- string是 C++ 特有的类,输入输出string对象需使用cin和cout。
- 使用cin和cout需包含头文件iostream

#include <iostream>

并在代码开头声明名称空间std;

using namespace std;



■ cin/cout示例

Code

```
int a;
char s[10];
string str;
cin >> a;
cin >> s >> str;
cout << a << ' ' << s;</pre>
cout << str << endl;</pre>
//<<和>>可以连接使用
//<<和>>形象表示数据的流向
//endl相当于'\n'
```

Input

123 Hello String

Output

123 HelloString

字符串与数字的转换

C

```
//字符与数字
                // 字符串与数字
char a = '8';
              char str[10] = "1234";
                int num=atoi(str); //使用 atoi函数
int b = a-'0';
                 char result[10]={0};
int b = 9;
                 num=1234;
a = b + '0';
                 itoa(num, result, 10); //使用itoa函数,10进制
                 char str[10] = \{0\};
                 sprintf(str, "%d", num); // 使用sprintf函数
                 long long c = 0;
                 char str1[10] = "12345678";
                 sscanf(str1, "%d", &c); //字符串转数字
```

C++

```
string s("12345");
long long x = stoi(s);
cout << x << endl;
long long m = 1234567890;
string str2 = to_string(m); //数字转字符
cout << str2 << endl;
```

建议设计封装自己的转换函数

字符串处理示例

Description

在线评测系统读取正确的输出和用户的输出进行对比:

- 如果完全相同,输出 "Accepted"
- 如果只有空格、Tab 或者回车不同,输出 "Presentation Error"
- 否则,输出"Wrong Answer"

现给你一些正确输出和用户输出,请你给出评测机正确的输出。

Input

$$1 + 2 = 3$$
 $1+2=3$

Output

Presentation Error



Solution

开始标记为 "AC" 状态,同时扫描两个串

如果遇到两个字符不同

如果是空白符,则标记为 "PE" 状态,跳过,继续扫描

如果都不是空白符,标记为 "WA" 状态,停止扫描

输出当前状态

Input

$$1 + 2 = 3$$

 $1+2=3$

建议自己实现