

# NOI(全国青少年信息学奥林匹克竞赛)

## 第四课 - 字符类型和字符数组

### 字符类型

```
//定义一个字符变量
```

```
char a;
```

有什么特性？占一个字节，可以进行类似int的加减操作

单引号括起来的单个字符就是字符常量，比如 'a'

### 字符数组

```
//定义一个字符数组
```

```
char a[5]; //数组a是一个具有5个字符元素的一维数组
```

```
char b[3][5]; //数组b是一个二维数组
```

```
//字符数组的赋值：
```

```
// 1. 字符数组的初始化
```

```
char a[5] = {'a', 'b', 'c', 'd', 'e'};
```

```
// 2. 数组元素的赋值
```

```
char a[3];
```

```
a[0] = 'a';
```

```
a[1] = 'b';
```

```
a[2] = 'c';
```

### 字符串

字符串就是一个字符数组，但它规定了要以字符 '\0' 作为这个字符数组的结束标志，注意，这个 '\0' 也会占一个字符的存储空间。

双引号括起来的是字符串常量，比如 "abcd"

猜猜"abcd"占几个字节？

```
// 初始化字符串变量
char b[5] = "abcd"; //双引号括起来的是字符串常量，b就是字符串
char b[5] = {'a', 'b', 'c', 'd', '\0'}; //等价于上面的写法

int size = sizeof(b); //返回5，表示数组它占了5个字节的存储空间
int len = strlen(b); //返回4，，表示这个字符串b的实际长度是4，它会从数组的首地址开始一直找下去，直到遇到'\0'停止查找

char b[5] = "abcde"; //错误，数组空间不够
```

## 输入与输出

```
// 输入
char c;
c = getchar(); //可以读取空格，需要#include <string>

c = cin.get(); //cin.get可以读取任意字符包括空格、换行符等
cin.get(c); //同上

char a[10];
scanf("%s", a); //读取输入时会忽略开头的空格，然后开始读取，遇到换行符停止，并且读取过程中一旦遇到空格，则空格之后的都会忽略。

char b[10];
cin.getline(b, 10); //读取一个字符串（字符数组的最后一位是 '\0'，strlen返回9），接受空格，默认是遇到换行符停止（并且在缓冲区里面删除掉换行符）

string s;
cin >> s; //cin在读入时会忽略开头的空格、tab键、换行符（就是回车键），遇到空格、换行符就停止

getline(cin, s); //可以接收空格、tab键、换行符，需要#include <string>，遇到换行停止（string s中包含了最后的换行符）

char b[10];
gets(b); //接受空格，遇到回车时截止（如果输入超出数组长度，会修改其他的变量，不建议使用）

// 输出
cout << a;
printf("%s", a);
```

## 练习

1. 输入两行内容，第一行是一段不超过100个字符的字符串，第二行包含以空格分隔的两个字符A和B，请将第一行中所有的字符A替换成字符B后输出。

## 字符串处理函数 -- 字符数组相关

```
// 需要cstring头文件
#include <cstring>
```

```

char a[100] = "abcd";
int size = sizeof(a); // 返回数组的长度
int len = strlen(a); //返回字符串的长度（实际长度）

char b[100];
strcpy(b, a); //复制字符串a到字符串b，返回字符串b的值，注意请确保数组b的长度是足够的
strncpy(b, a, 100); // 用法同上，第三个参数指定了只复制字符串a前n个字符

strcat(a, b); // append b to a, where its terminating null character is
overwritten by the first character of b.
strstr(a, b); //如果b不是a的子串则返回NULL，否则返回a中b首次出现的地址

int s = strcmp(a, b); /*比较字符串a和b的大小，
如果a > b，返回一个正整数；
如果a == b，返回0；
如果a < b，返回一个负整数*/

strlwr(a); //大写字母换成小写字母
strupr(a); //小写字母换成大写字母

char c;
char d = toupper(c); //字符转换为大写

// refer to http://www.cplusplus.com/

```

## 练习

- 判断两个由大小写字母和空格组成的字符串在忽略大小写，且忽略空格后是否相等

【输入】

两行，每行包含一个字符串

【输出】

若两个字符串相等，输出YES 否则输出NO

- 输入两个字符串，要求判定其中一个字符串（短的）是否可以通过若干次的循环移位变成另一个字符串的子串。

【输入】

一行，包含两个字符串，中间由单个空格隔开。字符串只包含字母，长度不超过30.

【输出】

如果判定成功，输出true，否则输出false

## 字符串string类型

在运用字符数组时，我们不能两个字符串进行赋值操作，比如：

```

char a[5] = "abcd";
char b[5] = a; // 错误，无法直接赋值

```

利用C++强大的string类型，我们可以摆脱字符数组，进行更方便的字符串操作：

```

#include <string>
using namespace std;
int main() {
    string str1; //定义一个字符串，初始化为空
    string str2("word"); //用字符串常量"word"初始化一个字符串
    string str3(str2); //用一个字符串初始化另一个字符串
    string str4 = str3; //一个字符串赋值给另一个字符串

    int len = str4.length(); //返回字符串的字节长度
    int size = str4.size(); //同上
    string str5 = "hello" + str4; //字符串的连接直接用加号，str5现在是什么

    char c = str5[3]; //返回的c是什么？

    //获取子字符串
    /*
    s.substr(pos, n) 返回一个string，包含s中从pos开始的n个字符的拷贝（pos的默认值是0，n的
    默认值是s.size() - pos，即不加参数会默认拷贝整个s）
    */
    string substr = str5.substr(0, 5); // 返回的substr是什么？

    //替换 replace()函数
    string ss="abcdefg";
    ss.replace(2,2,"999"); //ss为ab999efg

    //查找
    int i = ss.find("efg"); // 查得到则返回目标的第一个下标，查不到返回ss.npos

    //字符串转数字
    /*
    int stoi (const string& str, size_t* idx = 0, int base = 10);
    将字符串 str 按照指定的基数 base 解释为 int 类型的整数。
    如果idx不是空指针，则该函数还将idx的值设置为该数字后str中第一个字符的位置。
    如果将base设为0，则该函数将根据 str 的格式来自动决定基数。
    */

    std::string str_dec = "2001, A Space Odyssey";
    std::string str_hex = "40c3";
    std::string str_bin = "-10010110001";
    std::string str_auto = "0x7f";

    std::string::size_type sz; // alias of size_t

    int i_dec = std::stoi (str_dec,&sz);
    int i_hex = std::stoi (str_hex,nullptr,16);
    int i_bin = std::stoi (str_bin,nullptr,2);
    int i_auto = std::stoi (str_auto,nullptr,0);

    std::cout << str_dec << ": " << i_dec << " and [" << str_dec.substr(sz) <<
    "]\n";
    std::cout << str_hex << ": " << i_hex << '\n';
    std::cout << str_bin << ": " << i_bin << '\n';
    std::cout << str_auto << ": " << i_auto << '\n';

    //数字转字符串
    /*

```

```

string to_string (int val); //属于C++11, 通过添加gcc编译选项 -std=c++11

sprintf(s, "%d", N); //将整数N转换为字符串s
*/
string s = to_string(5);
sprintf(s, "%d", 5);

}

// stoi example
#include <iostream>    // std::cout
#include <string>      // std::string, std::stoi

int main () {
    std::string str_dec = "2001, A Space Odyssey";
    std::string str_hex = "40c3";
    std::string str_bin = "-10010110001";
    std::string str_auto = "0x7f";

    std::string::size_type sz;    // alias of size_t

    int i_dec = std::stoi (str_dec,&sz);
    int i_hex = std::stoi (str_hex,nullptr,16);
    int i_bin = std::stoi (str_bin,nullptr,2);
    int i_auto = std::stoi (str_auto,nullptr,0);

    std::cout << str_dec << ": " << i_dec << " and [" << str_dec.substr(sz) <<
    "]\n";
    std::cout << str_hex << ": " << i_hex << '\n';
    std::cout << str_bin << ": " << i_bin << '\n';
    std::cout << str_auto << ": " << i_auto << '\n';

    return 0;
}

```

## 建议

所有字符串相关的操作，尽量使用string类型; 如果要用字符数组，记住要初始化，比如char a[10] = {\0};

## 课后习题

### 1130：找第一个只出现一次的字符

[http://ybt.ssoier.cn:8088/problem\\_show.php?pid=1130](http://ybt.ssoier.cn:8088/problem_show.php?pid=1130)

#### 【题目描述】

给定一个只包含小写字母的字符串，请你找到第一个仅出现一次的字符。如果没有，输出no。

#### 【输入】

一个字符串，长度小于100000。

#### 【输出】

输出第一个仅出现一次的字符，若没有则输出no。

【输入样例】

abcabd

【输出样例】

c

### 1129 : 统计数字字符个数

[http://ybt.ssoier.cn:8088/problem\\_show.php?pid=1129](http://ybt.ssoier.cn:8088/problem_show.php?pid=1129)

【题目描述】

输入一行字符，统计出其中数字字符的个数。

【输入】

一行字符串，总长度不超过255。

【输出】

输出为1行，输出字符串里面数字字符的个数。

【输入样例】

Peking University is set up at 1898.

【输出样例】

4

### 1138 : 将字符串中的小写字母转换成大写字母

【题目描述】

给定一个字符串，将其中所有的小写字母转换成大写字母。

【输入】

输入一行，包含一个字符串（长度不超过100，可能包含空格）。

【输出】

输出转换后的字符串。

【输入样例】

helloworld123Ha

【输出样例】

HELLOWORLD123HA

### 1136 : 密码翻译

【题目描述】

在情报传递过程中，为了防止情报被截获，往往需要对情报用一定的方式加密，简单的加密算法虽然不足以完全避免情报被破译，但仍然能防止情报被轻易的识别。我们给出一种最简的的加密方法，对给定的一个字符串，把其中从a-y，A-Y的字母用其后继字母替代，把z和Z用a和A替代，其他非字母字符不变，则可得到一个简单的加密字符串。

【输入】

输入一行，包含一个字符串，长度小于80个字符。

#### 【输出】

输出每行字符串的加密字符串。

#### 【输入样例】

Hello! How are you!

#### 【输出样例】

lfmmp! lpx bsf zpv!

## 总结

---

C++头文件:

```
#include
```

```
#include
```

c头文件:

```
#include <string.h>
```

```
#include <stdio.h>
```

本节介绍了很多常用的函数，为下节课的函数做铺垫，介绍如何使用 [www.cplusplus.com](http://www.cplusplus.com)

课堂的练习通过github classroom上传，课后老师会做批改

## 课堂总结回顾

---

1. 复习二维数组的课后作业
2. 字符类型和字符数组
3. 什么是字符串（后面包含'\0'）
4. 常用的输入控制语句 cin.getline(), getchar()
5. 常用的字符串操作函数（拷贝strcpy, 添加strcat, 比较strcmp, 查找strstr, 大小写转换strlwr,strupr）

#### 一些问题：

1. 有些同学在写if语句的时候，会用错=和==
2. 有些同学会在定义数组的时候，用一个变量, 比如 char s[n];
3. 判断字符不相等，我们要使用 !=

对于字符和字符串类型的输入，有很多选择，c风格的scanf，c++的cin.getline, getchar，gets等等，我们现在**统一标准**：

1. 对于需要读取包含空格的字符串：cin.getline(s, 100);（第二个参数一般设置为字符数组s的大小，表示最多会读99个字符放到数组中）
2. 对于需要读取不包含空格的字符串：cin >> s;（这里的s可以是字符数组，也可以是string类型）
3. 对于需要读取单个字符（注意：空格也是单个字符）：char c = getchar();
4. 对于需要读取多个字符（在同一行中输入，每个字符之间可能有空格分隔）：cin >> c1 >> c2 >> c3;