集训课-13：字符串与字符串数组

课程目标

01 字符数组与字符串

02 字符数组函数

03 string与string函数

04 string数组

05 总结回顾

Part 1

字符数组与字符串

字符数组与字符串

字符 ‘h’ ‘e’ ‘l’ ‘l’ ‘o’

字符串 “hello”

字符数组与字符串

字符：单引号引起的单个内容

字符串：双引号引起来的一个或者一串内容

字符数组与字符串

字符数组可以表示字符串

cin >> 字符数组名;

cout << 字符数组名;

字符数组与字符串

cin >> 字符数组名;

cout << 字符数组名;

只输出第一个空格之前的内容

字符数组与字符串

getline函数可以从输入流中读取一行字符，其中包含空格。

cin.getline(数组名,字符数组长度);

得到行

字符数组与字符串

使用getline函数时必须要加#include <cstring>

```
#include <cstring>
……
char a[10];
cin.getline(a,10);
cout << a;
```

字符数组与字符串

将字符串存储在字符数组中：

存储字符串时，数组长度至少多一个。

下标	0	1	2	3	4	5
内容	'h'	'e'	'l'	'l'	'o'	'\0'

字符数组与字符串

将字符串“hello“存储在字符数组中：

方法1： `char a[5] = {'h','e','l','l','o'};`

方法2： `char a[6] = "hello";`

Part 2

字符数组函数

字符数组函数

strlen函数的返回值为整型数据，表示字符串中有效字符的个数。

strlen(字符数组名);

字符数组函数

使用strlen函数可以直接统计字符串长度。

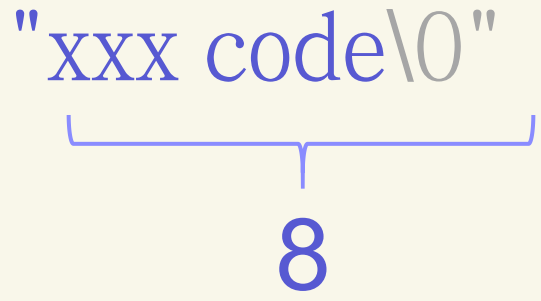
```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    char a[100];
    cin.getline(a,100);
    cout << strlen(a);
    return 0;
}
```

字符数组函数

strlen函数的返回值为整型数据，表示字符串中有效字符的个数



✓字母

✓数字

✓空格

✓特殊字符

☒ '\0'

字符数组函数

strlen函数只统计'\0'前所有字符数量

```
char a[100] = {'t','a','3','\0','w','\0'};
cout << strlen(a);
```

* 如果字符串中含有多个'\0'，只统计第一个'\0'前面有效字符个数。

字符数组函数

strcmp函数可以比较两个字符串的大小

```
#include <cstring>
      ……
char a[100] = "IronMan";
char b[100] = "Hulk";
cout  << strcmp(a,b);
```

字符数组函数

比较大小cmp，正大负小双等于

```
#include <cstring>
        ……
char a[100] = "xes";
char b[100] = "code";
cout  << strcmp(a,b);
```

① 如果返回正数， 字符串a > b

② 如果返回负数， 字符串a < b

③ 如果返回数字0 ， 字符串a == b

字符数组函数

strcpy函数是专门用来拷贝字符串的函数。

strcpy(字符数组名1,字符数组名2);

字符数组函数

字符串b的空间要足够大，执行strcpy操作后，会将整个a字符串拷贝给b字符串，包括'\0'。

```
#include<cstring>
……
char a[100] = "helloworld";
char b[100];
strcpy(b,a);
cout << b;
```

字符数组函数

常用函数	功能
getline(a,10)	读入一行含空格的字符串数据
strlen(a)	获取字符串长度 包含空格和其他字符，不包含'\0'
strcmp(a,b)	比较两个字符串的大小 从左至右，依次比较每个字符的ASCII码值
strcpy(a,b)	将字符串b拷贝给字符串a 字符串a的空间要足够大

Part 3

string与string函数

string与string函数

string字符串的定义

⭐ string 变量名;

```
string password;
```

注意： #include <string>

string字符串初始化

初始化空的string字符串

```
string password;
```

string与string函数

string字符串输出

- 格式：cout << 变量名;

```
string password = "Aa!12";
cout << password;
```

string与string函数

string字符串不带空格输入

★ 格式：cin >> 变量名;

```
string password;
cin >> password;
```

string与string函数

string字符串带空格输入

⭐ 格式：getline(cin, 变量名);

```
string password;
getline(cin, password);
```

string与string函数

string字符串拼接

⭐ 格式：变量名1 + 变量名2;

```
string birth = "20020401";
string name = "kevin";
password = birth + name;
```

string与string函数

string字符串比较

⭐ 从前到后逐位比较，一旦不同，则根据ASCII码比较大小

```
string a = "abc", b = "abb";

string a = "d", b = "abcd";
```

string与string函数

string字符串比较

- 格式：变量名1 关系运算符 变量名2

```
string word1 = "abc";
string word2 = "abb";
if (word1 == word2)
{
    cout << "yes";
}
```

string与string函数

string中元素获取

- 通过下标获取字符串元素的值

格式：变量名[下标];

```
string password = "hello world";
cout << password[0];
```

string与string函数

string中元素获取

遍历输出时，i < 字符串长度

```
string password = "helloss";
for (int i = 0; i < 7; i++)
{
    cout << password[i] ;
}
```

string与string函数

获取string长度

⭐ 格式：变量名.size();

```
string password = "hellosahdkjxZl1JKLHKj1hlkaslj";
cout << password.size();
```

string与string函数

string查找

find()函数

在一个字符串中查找另一个字符串或单个字符

字符串名称.find（目标字符串或字符）；

string与string函数

string查找

必须使用 #include <string>，查找字符串用双引号，返回目标字符串首字母在原串中的位置下标

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string a = "VIPBG";
    cout << a.find("VIP");
    return 0;
}
```

string与string函数

未找到

string::npos

no position

无效位置

a.find("BBQ") == string::npos

表示在字符串a中不存在"BBQ"

string与string函数

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    //在字符串a中寻找字符串b
    string a,b;
    getline(cin,a);
    getline(cin,b);
    if (a.find(b) == string::npos) //如果没有目标字符串
    {
        cout << "NO"; //输出NO
    }
    else //有目标字符串
    {
        cout << a.find(b); //输出字符串首字母首次出现的位置
    }

    return 0;
}
```

string与string函数

string截取

substr()

截取一个字符串的子串

a.substr(开始下标,截取长度);

string与string函数

string截取

a.substr(3,4);

string与string函数

在字符串a中，从下标n开始，截取长度为m的子串

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string a;
    getline(cin,a);
    int n,m;
    cin >> n >> m;
    cout << a.substr(n,m);
    return 0;
}
```

string与string函数

string反转

reverse()函数

将字符串中的字符按顺序反转

F E D C B A 反转前

A B C D E F 反转后

string与string函数

string反转

reverse()函数

① 必须使用 #include <algorithm>

algorithm

算法

string与string函数

string反转

reverse()函数

② 从头到尾，一点到头，一点到尾

reverse(字符串名称.begin(),字符串名称.end());

reverse(a.begin(),a.end());

string与string函数

判断回文数

```
#include <iostream>
#include <string>
#include <algorithm> //算法头文件
using namespace std;

int main()
{
    string str1,str2;
    getline(cin,str1);
    str2 = str1;
    reverse(str2.begin(),str2.end()); //字符串反转
    if (str2 == str1) //判断是否相等
    {
        cout << "yes";
    }
    else
    {
        cout << "no";
    }
    return 0;
}
```

Part 4

string数组

string数组

可以使用string数组存储多个字符串

string 数组名[字符串个数];

string name[5];

string数组

可以使用字符串的形式进行初始化

```
string name[5] = { "hello",
                   "nihao",
                   "xxcode",
                    “xixi",
                   "nicai"
                    };
```

string数组

数组的行下标从0开始

```
string name[5] = { "hello",   //0
                   "nihao",   //1
                   "xxxcode", //2
                    “xixi",   //3
                   "nicai"    //4
                    };
```

string数组

可以使用for循环输出

```
string name[5]={"hello","nihao","xxcode",
                “xixi","nicai"};

for(int i = 0; i < 5; i++)
{
        cout << name[i] <<endl;
}
```

string数组

可以使用for循环输入

```
string name[100] = {};
int n;
cin >> n;

for (int i = 0; i < n; i++)
{
    cin >> name[i];
}
```

Part 5

总结回顾

总结回顾

	字符数组	string字符串
定义	char a[10], b[10];	string a, b;
头文件	#include <cstring>	#include <string>
不含空格输入	cin >> a;	cin >> a;
含空格输入	cin.getline(a,10);	getline(cin,a);
直接输出	cout << a;	cout << a;
输出某一个元素	cout << a[下标];	cout << a[下标];

遍历输出	for (int i = 0; i < 10; i++)
	{
	cout << a[i];
	}
		for (int i = 0; i < 10; i++)
		{
		cout << a[i];
		}

总结回顾

	字符数组	string字符串
获取长度	strlen(a);	a.size();

比较	strcmp(a,b);
	//返回值为0， a等于b
	//返回值大于0， a大于b
	//返回值小于0， a小于b
		a > b
		a == b
		a < b

拼接	strcat(a,b);
	//b拼接到a的末尾
		string c;
		c = a + b;
		//a和b进行拼接

拷贝	strcpy(b,a);
	//把a拷贝给b
		string a = “abc”, b;
		b = a;
		//把a拷贝给b

总结回顾

获取string长度

⭐ 格式：变量名.size();

```
string password = "hellosahdkjxZl1JKLHKj1hlkaslj";
cout << password.size();
```

总结回顾

查找字符串

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    //在字符串a中寻找字符串b
    string a,b;
    getline(cin,a);
    getline(cin,b);
    if (a.find(b) == string::npos) //如果没有目标字符串
    {
        cout << "NO"; //输出NO
    }
    else //有目标字符串
    {
        cout << a.find(b); //输出字符串首字母首次出现的位置
    }

    return 0;
}
```

总结回顾

在字符串a中，从下标n开始，截取长度为m的子串

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string a;
    getline(cin,a);
    int n,m;
    cin >> n >> m;
    cout << a.substr(n,m);
    return 0;
}
```

总结回顾

判断回文数

```
#include <iostream>
#include <string>
#include <algorithm> //算法头文件
using namespace std;

int main()
{
    string str1,str2;
    getline(cin,str1);
    str2 = str1;
    reverse(str2.begin(),str2.end()); //字符串反转
    if (str2 == str1) //判断是否相等
    {
        cout << "yes";
    }
    else
    {
        cout << "no";
    }
    return 0;
}
```

Class is over