《计算概论A》 程序设计部分

字符数组与字符串

李 戈

北京大学 信息科学技术学院 软件研究所

lige@sei.pku.edu.cn

字符数组的定义

```
#include<iostream>
using namespace std;
int main()
{
        char a[10] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j' };
        for (int i = 0; i < 10; i++)
                cout << a[i];
        return 0;
}
```

a	b	c	d	e	f	g	h	i	j
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

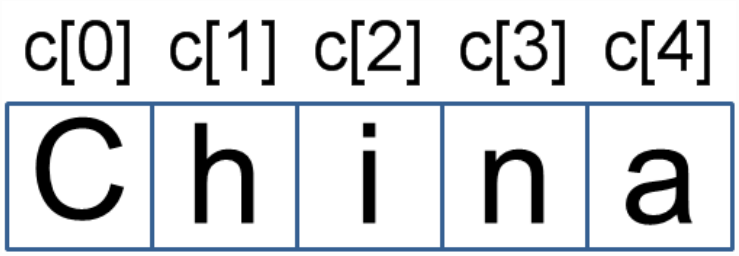
字符数组的定义

```
#include<iostream>
using namespace std;
int main()
{
        char a[10] = { 'a', 'b', 'c', 'd', 'e'};
        for (int i = 0; i < 10; i++)
                cout << a[i];
        return 0;
}
```

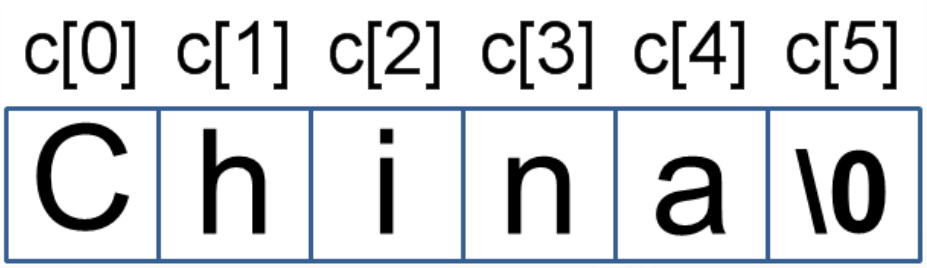
a	b	c	d	e	\0	\0	\0	\0	\0
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

字符数组的初始化

char c[ ] = {‘C', ‘h ', ‘i', ‘n', ‘a'};



char c[ ] = “China”;



认识一下字符串

“China”;

char c[5] = “China”; ?

关于赋值

- **只可以：**
  - 在数组定义并初始化的时候：

char c[6] = “China”;

- **不可以：**
  - 不能用赋值语句将一个字符串常量或字符数组直接赋给另一个字符数组。

str1[ ]= “China”; （赋值，不合法）

str1 = “China”; （赋值，不合法）

str2 = str1; （赋值，不合法）

正确的赋值方式

```
#include<iostream>
using namespace std;
int main()
{
        char str1[] = "C++ language", str2[20];
        int i = 0;
        while (str1[i] != '\0')
        {
                str2[i] = str1[i];
                i++;
        }
        str2[i] = '\0';
        cout << "String1:" << str1 << endl;
        cout << "String2:" << str2 << endl;
        return 0;
}
```

字符串数组

- 利用二维数组存储多个字符串

char weekday[7][11] = {“Sunday", "Monday", "Tuesday", "Wednesday", “Thursday", “Friday", “Saturday"};

S	u	n	d	a	y	\0	\0	\0	\0	\0
M	o	n	d	a	y	\0	\0	\0	\0	\0
T	u	e	s	d	a	y	\0	\0	\0	\0
W	e	d	n	e	s	d	a	y	\0	\0
T	h	u	r	s	d	a	y	\0	\0	\0
F	r	i	d	a	y	\0	\0	\0	\0	\0
S	a	t	u	r	d	a	y	\0	\0	\0

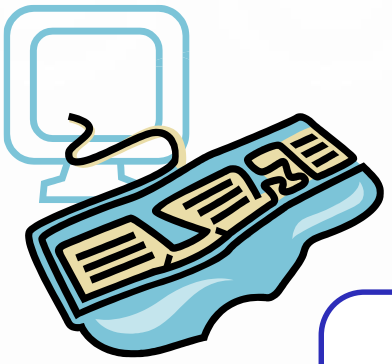
字符/字符数组/字符串的输入与输出

I. 引子：输入的过程

II. 一个字符的输入与输出

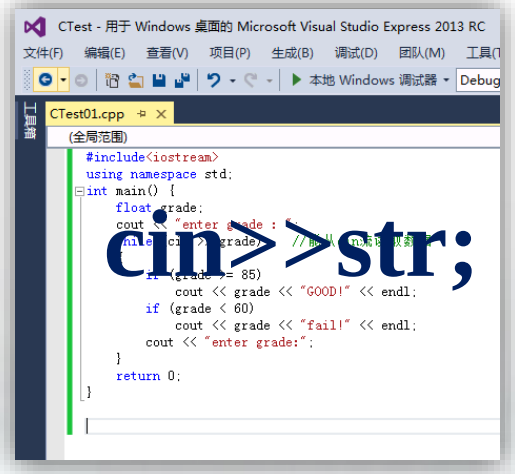
III. 字符数组/字符串的输入与输出

认识一下：输入缓冲区



How are you ?

How are you?↵



用cin输入数据

```
int a, b ;
cin>>a>>b; // 从键盘输入 21  22↙
```

```
int a, b ;
cin>>a>>b; // 从键盘输入 21  abc ↙
```

```
int a, b, c ;
cin>>a>>b>>c;   // 从键盘输入 21  22 ↙
                // 23 ↙
```

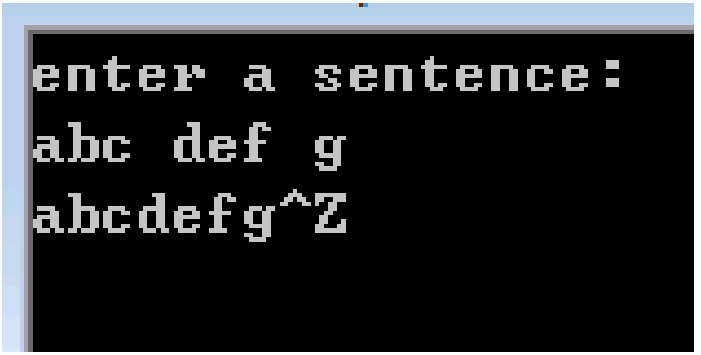
用cin输入数据

```
#include<iostream>
using namespace std;
int main() {
        float grade;
        cout << "enter grade : ";
        while (cin >> grade)      //能从cin流读取数据
        {
                if (grade >= 85)
                        cout << grade << "GOOD!" << endl;
                if (grade < 60)
                        cout << grade << "fail!" << endl;
                cout << "enter grade:";
        }
        return 0;
}
```

（1）一个字符的输入

方法一：直接用cin输入字符

```
#include <iostream>
using namespace std;
int main()
{
	char c;
	cout << "enter a sentence:" << endl;
	while (cin >> c)
		cout << c;
	return 0;
}
```

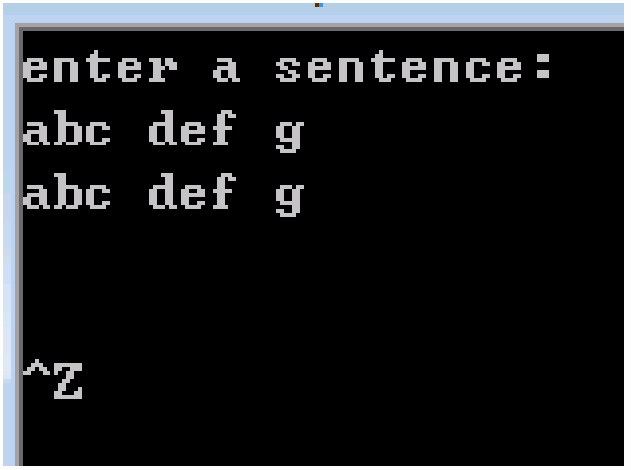


用 cin.get( ) 函数输入字符

- cin.get( )函数
  - 可以用于读入一个字符；
  - 2种形式: 无参数，一个参数。
    - cin.get( )
    - cin.get(char)

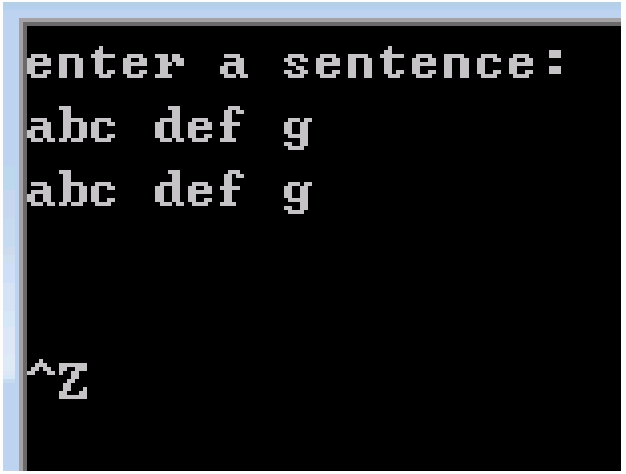
方法二：用 cin.get( ) 输入字符

```
#include <iostream>
using namespace std;
int main()
{
        char c;
        cout << "enter a sentence:" << endl;
        while ((c = cin.get()) != EOF)
                cout<<c;
        return 0;
}
```



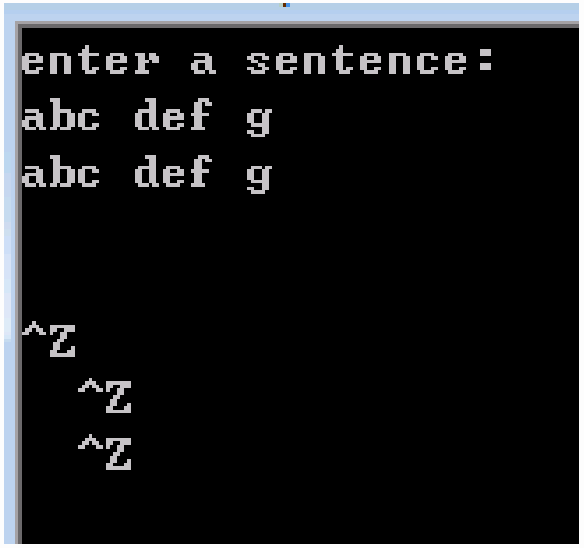
方法三：用 cin.get(char) 输入字符

```
#include <iostream>
using namespace std;
int main()
{
        char c;
        cout << "enter a sentence:" << endl;
        while (cin.get(c))
                        //读取一个字符赋给字符变量c
                cout << c;
        return 0;
}
```



方法四：用 getchar( ) 输入字符

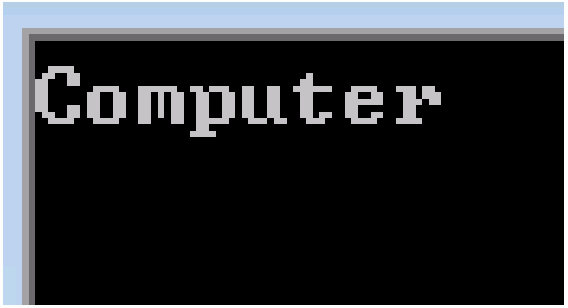
```
#include <iostream>
using namespace std;
int main()
{
        char c;
        cout << "enter a sentence:" << endl;
        while (c = getchar()) //不跳过任何字符
                cout<<c;
        return 0;
}
```



（2）一串字符的输出

用 cout 输出字符数组

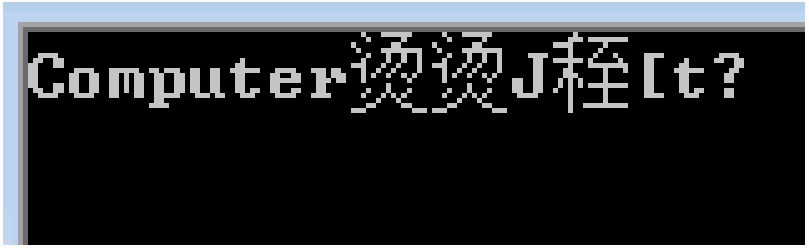
```
#include <iostream>
using namespace std;
int main()
{
        char a[10] = "Computer";
        cout<<a;
        return 0;
}
```



C	o	m	p	u	t	e	r	\0	\0
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

用 cout 输出字符数组

```
#include <iostream>
using namespace std;
int main()
{
        char a[8] = { 'C', 'o', 'm', 'p', 'u', 't', 'e', 'r' };
        cout<<a;
        return 0;
}
```



C	o	m	p	u	t	e	r
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]

用 cout 输出字符数组

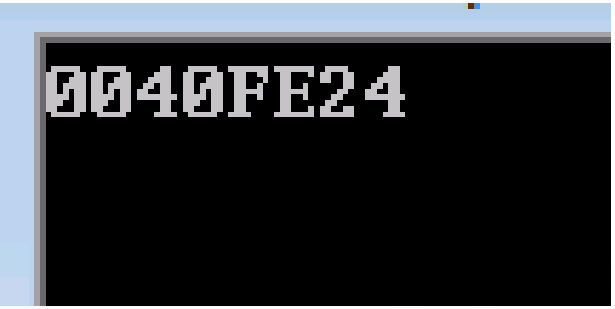
```
#include <iostream>
using namespace std;
int main()
{
        char weekday[7][11] = { "Sunday", "Monday",
"Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday" };
        for (int i = 0; i < 7; i++)
                cout << weekday[i] << endl;
        return 0;
}
```

S	u	n	d	a	y	\0	\0	\0	\0	\0
M	o	n	d	a	y	\0	\0	\0	\0	\0
T	u	e	s	d	a	y	\0	\0	\0	\0
W	e	d	n	e	s	d	a	y	\0	\0
T	h	u	r	s	d	a	y	\0	\0	\0
F	r	i	d	a	y	\0	\0	\0	\0	\0
S	a	t	u	r	d	a	y	\0	\0	\0

```
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

用 cout 输出字符数组

```
#include <iostream>
using namespace std;
int main()
{
        int a[8] = { 1,2,3,4,5,6 };
        cout<<a;
        return 0;
}
```



（2）一串字符的输入

方法一：直接用cin输入字符串

```
#include <iostream>
using namespace std;
int main()
{
        char str[10];
        cout << "enter a sentence:" << endl;
        while (cin >> str)
                cout << str<<endl;
        return 0;
}
```

```
enter a sentence:
abc def g
abc
def
g
^Z
```

方法二：用cin.get( )函数输入

- 有3个参数的get函数

```
cin.get(ch, 10,'\n');
```

  - 读取10-1个字符（包含空格），赋给指定的字符数组;
  - 如果在读取10-1个字符之前，遇到指定的终止字符’\n’，则提前结束读取;（如果第3个参数没有指定，则默认为'\n’
  - 读取成功返回非0值(真)，如失败(遇文件结束符)则返回0值(假)。

方法二：用cin.get( )函数输入

```
#include <iostream>
using namespace std;
int main()
{
        char ch[20];
        cout << "enter a sentence:" << endl;
        cin.get(ch, 10, ‘o’); //指定终止符为’o’
        cout << ch << endl;
        return 0;
}
```

```
enter a sentence:
We are good friends.
We are g
```

方法三：用cin.getline( )函数输入

```
#include <iostream>
using namespace std;
int main()
{
        char ch[20];
        cout << "enter a sentence:" << endl;
        cin.getline(ch, 10, ‘o’); //指定终止符为’o’
        cout << ch << endl;
        return 0;
}
```

```
enter a sentence:
We are good friends.
We are g
```

方法三：用cin.getline( )函数输入

- getline与get的区别
  - getline遇到终止标志字符时结束，缓冲区指针移到终止标志字符之后；
  - get遇到终止字符是停止读取，指针不移动

We are good friends.

cin.get( )

We are good friends.

cin.getline( )

方法三：用cin.getline( )函数输入

```
#include <iostream>
using namespace std;
int main()
{
        char weekday[7][11];
        for (int i = 0; i < 7; i++)
                cin.getline(weekday[i], 11);
        for (int i = 0; i < 7; i++)
                cout << weekday[i] << endl;
        return 0;
}
```

```
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

一个需要特别关注的程序

```
#include<iostream>
using namespace std;
int main()
{
        char a[10][10];
        int n = 0;
        cin >> n;
        for (int i = 0; i < n; i++)
                cin.getline(a[i], 10);
        for (int i = 0; i < n; i++)
                cout << a[i] << endl;
        return 0;
}
```

```
7
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday

Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
```

Pass !

```
#include<iostream>
using namespace std;
int main()
{
        char a[10][10];
        int n = 0;
        cin >> n;
        cin.get( ); //添加一条语句
        for (int i = 0; i < n; i++)
                cin.getline(a[i], 10);
        for (int i = 0; i < n; i++)
                cout << a[i] << endl;
        return 0;
}
```

例1 字符串加密

- 字符串加密
  - 输入一个字符串，把每个字符变成它后续字符，如果是‘Z’或者‘z’，则变成’A’或’a’.空格则不变。然后将变换后的字符串输出；
  - 要求能够接受连续输入；

```
hello
ifmmp
nice to meet u.
ojdf up nffu v/
do you like c++
ep zpv mjlf d,,
no?
op@
bye
czf
^Z
Press any key to continue
```

例1 字符串加密

- 思路：
  - 读入字符串（想一下以什么方式输入？）
  - 从字符头到尾循环：
    - 是'Z'则直接赋值'A'，跳过以下步骤
    - 是'z'则直接赋值'a'，跳过以下步骤
    - 空格不做处理，跳过以下步骤
    - 其他字符++
  - 输出新字符串；

例1 字符串加密

```
#include <iostream>
using namespace std;
int main()
{
        char str[200];
        while (cin.getline(str, 200))
        {
                for (int i = 0; str[i] != '\0'; i++)
                {
                        if (str[i] == 'Z')
                        {
                                str[i] = 'A';  continue;
                        }
                        if (str[i] == 'z')
                        {
                                str[i] = 'a';  continue;
                        }
                        if (str[i] == ' ')
                                continue;
                        str[i]++;
                }
                cout << str << endl;
        }
        return 0;
}
```

```
hello
ifmmp
nice to meet u.
ojdf up nffu v/
do you like c++
ep zpv mjlf d,,
no?
op@
bye
czf
^Z
```

例2 字符串连接

- 问题：
  - 输入两个字符串，将其中较短的串接到较长的串的后面。
- 要求：
  - 不使用系统函数 strcat
  - 每个输入的串的长度不超过20。

插入：字符数组常用操作

```
#include <iostream>
#include <string>
using namespace std;
int main()
{   char str1[20],str2[20];
    cin.getline(str1,20);
    strcpy(str2,str1);
    cout << str1 <<endl; //数组名就是字符串名
    cout << str2 <<endl;
}
```

例2 字符串连接

定义：

```
char str1[40],  str2[40];
cin.getline(str1,20);  cin.getline(str2,20);
```

计算长度：

```
for (len1 = 0; str1[len1] != '\0'; len1++);
for (len2 = 0; str2[len2] != '\0'; len2++);
```

拼接：第一个串的下标指向最后一个元素之后

第二个串的下标指向第一个元素。

```
for(len2=0;str1[len2]!='\0';len2++);
        str1[len1++]=str2[len2];
        str1[len1]='\0';     //不加上\0,就不是一个字符串。
```

```
#include <iostream>
using namespace std;
int main()
{
        int len1, len2; char str1[40], str2[40];
        cin.getline(str1, 20); cin.getline(str2, 20);
        for (len1 = 0; str1[len1] != '\0'; len1++);
        for (len2 = 0; str2[len2] != '\0'; len2++);
        if (len1 >= len2)
        {
                for (len2 = 0; str2[len2] != '\0'; len2++)
                        str1[len1++] = str2[len2];
                str1[len1] = '\0';
        }
        else
        {
                for (len1 = 0; str2[len1] != '\0'; len1++)
                        str2[len2++] = str1[len1];
                str2[len2] = '\0';
        }
        cout << str1 << endl;       cout << str2 << endl;
        return 0;
}
```

```
Computing
Introduction
Computing
IntroductionComputing
```

例3 统计单词数

- 问题：
  - 输入一个英文句子（不超过80个字母），统计其中有多少个单词，单词之间用空格分开。

例3 统计单词数

Everything Should Be Made as Simple as Possible, But Not Simpler.

- Albert Einstein

例3 统计单词数

```
#include <iostream>
using namespace std;
int main()
{
        char str[80];
        int num = 0, flag = 0;
        cin.getline(str, 80);
        for (int i = 0; str[i] != '\0'; i++)
        {
                if (str[i] == ' ')
                        flag = 0;
                else if (flag == 0)
                {
                        flag = 1; num++;
                }
        }
        cout << "字符串中有" << num << "个单词" << endl;
        return 0;
}
```

好好想想,有没有问题？

谢　谢！