



# 《计算概论A》课程 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](mailto:lige@sei.pku.edu.cn)



北京大学



# 字符数组的定义

## ■ 字符数组

◆ **char c[4]; char c[5][10];**

## ■ 字符数组的初始化

**char c[10] = {'c', ' ', 'P', 'r', 'o', 'g', 'r', 'a', 'm', 's'};**

```
#include<iostream>
using namespace std;
void main( )
{
    int i;  char a[10] ;
    for(i = 0; i < 8; i++)
    {
        a[i] = 'a';
    }
}
```



北京大学



# 字符数组的初始化

## ■ 定义(VC环境下)

```
char c[10] = {'c', ' ', 'P', 'r', 'o', 'g', 'r', 'a', 'm'};
```

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
c		p	r	o	g	r	a	m	\0

‘\0’代表ASCII码为0的字符, 不可显示, 是一个“空操作符”, 只作一个供辨别的标志。

## ■ 定义(VC环境下)

```
char c[10] = {'a', 'b', 'c'}
```

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



北京大学



# 字符数组的初始化

## ■ 定义

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

c[0]	c[1]	c[2]	c[3]	c[4]
C	h	i	n	a

## ■ 定义

`char c[ ] = {"China"};`

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]
C	h	i	n	a	\0



北京大学



# 字符数组与字符串

■ 定义: `char c[ ] = "China";`

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]
C	h	i	n	a	\0

■ 字符串:

◆ 如果一个字符数组的结束元素是‘\0’, 则可以作为字符串;

◆ `char c[5] = "China";` ?



北京大学



## 关于赋值

### ■ 只可以:

- ◆ 给一个字符变量或字符数组元素赋值

```
char a[5], c1, c2;          c1='A'; c2='B';
```

```
a[0]='C'; a[1]='h', a[2]='i'; a[3]='n'; a[4]='a';
```

### ■ 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。

```
char str1[ ]={"China"}, str2[20]; (初始化, 合法)
```

```
str1[ ]={"China"}; (赋值, 不合法)
```

```
str1 = "China"; (赋值, 不合法)
```

```
str2 = str1; (赋值, 不合法)
```



北京大学



## 正确的赋值方式

```
#include<iostream.h>
void 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;
}
```



北京大学





# 字符串数组

## ■ 利用二维数组存储多个字符串

(1) `char week[7][11]={"sunday","Monday","Tuesday","Wednesday","thursday","friday","saturday"};`

(2) `char grade[][8]`  
`={"excellent","good","middle","pass","bad"};`

(3) `char`  
`typename[6][10]={"int","double","char"};`

(4) `char d[10][20]={" "};`



北京大学





# 字符数组的输入

## ■ 字符数组的输入

### ◆ 整串输入的情况

`char str[15];`

`cin>>str;` //写字符串的名字, 不要下标

`how are you?` //系统将空格看作分隔符

`how '\0' → str;` //遇到空格结束

### ◆ 可以包含空格的输入

`cin.getline(str, 20);` //输入n-1个字符至str

输入为How are you? 则全部保存到str中。



北京大学



# 字符/字符数组/字符串 的输入与输出

引子：输入的过程

- 一个字符的输入与输出
- 字符数组/字符串的输入与输出



北京大学



# 输入缓冲区的概念

## ■ `cin>>a;`

- (1) 在用户输入完数据再按回车键后，该行数据被送入键盘缓冲区，形成输入流；
- (2) 流提取符“>>”从流中提取数据，这时流提取符通常跳过输入流中开始部分的空格、**tab**键、换行符等空白字符；
- (3) 程序中的变量通过流提取运算符“>>”从流中提取数据；
- (4) 遇到无效字符或文件结束标志，停止提取；



北京大学



## 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

```
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;  
}
```



北京大学



# (1) 一个字符的输入与输出



北京大学



## 例：用cin直接读入字符

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



北京大学





# 用get函数输入

## ■ get函数

- ◆ 用于读入一个字符或字符串；
- ◆ 共有3种形式: 无参数，一个参数，3个参数。

## ■ 不带参数的get函数cin.get( )

- ◆ 用来从指定的输入流中提取一个字符，函数的返回值就是读入的字符。



北京大学



## 例：用get函数读入字符

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



北京大学



# 用于字符输入的流成员函数

- 有一个参数的get函数cin.get(ch)
  - ◆ 从输入流中读取一个字符，赋给字符变量ch。
  - ◆ 如果读取成功则返回非0值(真)，如失败(遇文件结束符)则函数返回0值(假)。

```
#include <iostream>
using namespace std;
int main( )
{ char c;
  while(cin.get(c)) //读取一个字符赋给字符变量c,
  {                //如果读取成功, cin.get(c)为真
    cout.put(c);
  }
  return 0;
}
```



北京大学



## 例：用getchar函数读入字符

```
#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.put(c);
  return 0;
}
```



北京大学



## (2) 字符数组/字符串的输入





## 直接用cin输入字符串

### ■ 整串输入的情况

**char str[15];**

**cin>>str;**      //写字符串的名字, 不要下标

**how are you?** //系统将空格看作分隔符

**how '\0' → str;**      //遇到空格结束

```
#include <iostream>
using namespace std;
int main( )
{
    char str[10];
    if(cin>>str)
        cout<<str;
    return 0;
}
```



北京大学





## 用3个函数的get函数输入

### ■ 有3个参数的get函数

- ◆ **cin.get(字符数组, 字符个数n, 终止字符)**
- ◆ 从输入流中读取**n-1**个字符, 赋给指定的字符数组 (包含空格);
- ◆ 如果在读取**n-1**个字符之前遇到指定的终止字符, 则提前结束读取;
- ◆ 如果读取成功则函数返回非**0**值(真), 如失败(遇文件结束符) 则函数返回**0**值(假)。
- ◆ get函数中第3个参数可以省写, 此时默认为'\n'



北京大学





## 例：用get函数读入字符

```
#include <iostream>
using namespace std;
int main( )
{
    char ch[20];
    cout<<"enter a sentence:"<<endl;
    cin.get(ch, 10, '\n');    //指定换行符为终止字符
    cout<<ch<<endl;
    return 0;
}
```





# 用getline函数输入

## ■ 函数getline()

- ◆ 从输入流中读取一行字符
- ◆ 用法与带3个参数的get函数类似:

`cin.getline(字符数组(或字符指针), 字符个数n, 终止标志字符)`

## ■ 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



北京大学



## 例：用getline函数读入字符

```
#include <iostream>
using namespace std;
int main() {
    char ch[20];
    cout<<"enter a sentence:"<<endl;
    cin>>ch;
    cout<<"The string read with cin is:"<<ch<<endl;
    cin.getline(ch, 20, '/');    //读个字符或遇'/'结束
    cout<<"The second part is:"<<ch<<endl;
    cin.getline(ch, 20);          //读个字符或遇'/n'结束
    cout<<"The third part is:"<<ch<<endl;
    return 0;
}
```

若输入： I like C++./I study C++./I am happy.✓

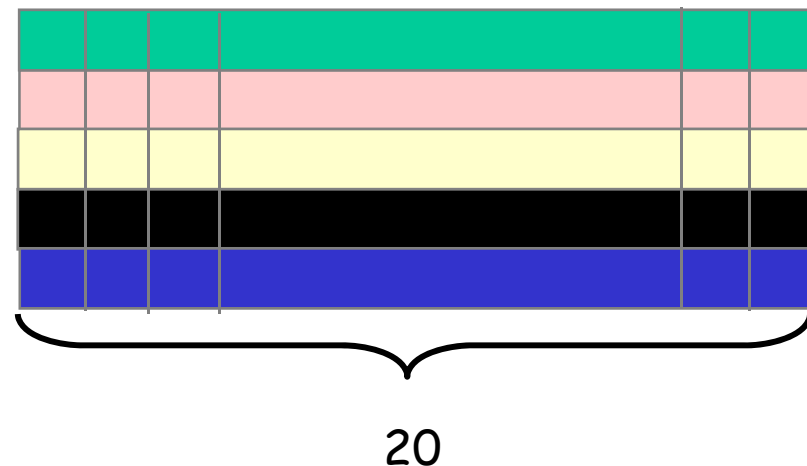


北京大学

# 字符数组常用操作

- 输入多个字符串的情况

```
int main()
{ int i;
  char str[5][20];
  for(i=0;i<5;i++)
    cin.getline(str[i], 20);
  for(i=0;i<5;i++)
    cout<<str[i];
  return 0;
}
```



北京大学



## 这个程序无法通过Grids，为什么？

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



北京大学



## 修改一下

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



北京大学





# 字符数组举例 (1)

## ■ 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
```



北京大学





## 字符数组举例（1）

### ■ 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；



北京大学

```
#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;
}
```



## 字符数组举例 (2)

### ■ 问题:

- ◆ 输入两个字符串，将其中较短的串一个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。



北京大学



## 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;
```

```
}
```



北京大学



## 字符数组举例 (2)

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;str2[len2]!='\0';len2++);
```

```
str1[len1++]=str2[len2];
```

```
str1[len1]='\0'; //不加上\0,就不是一个字符串。
```



北京大学

```
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; }
```



## 字符数组举例 (3)

### ■ 问题:

- ◆ 输入一个英文句子，统计其中有多少个单词，单词之间用空格分开。



北京大学





## 字符数组举例 (3)

### ■ 分析:

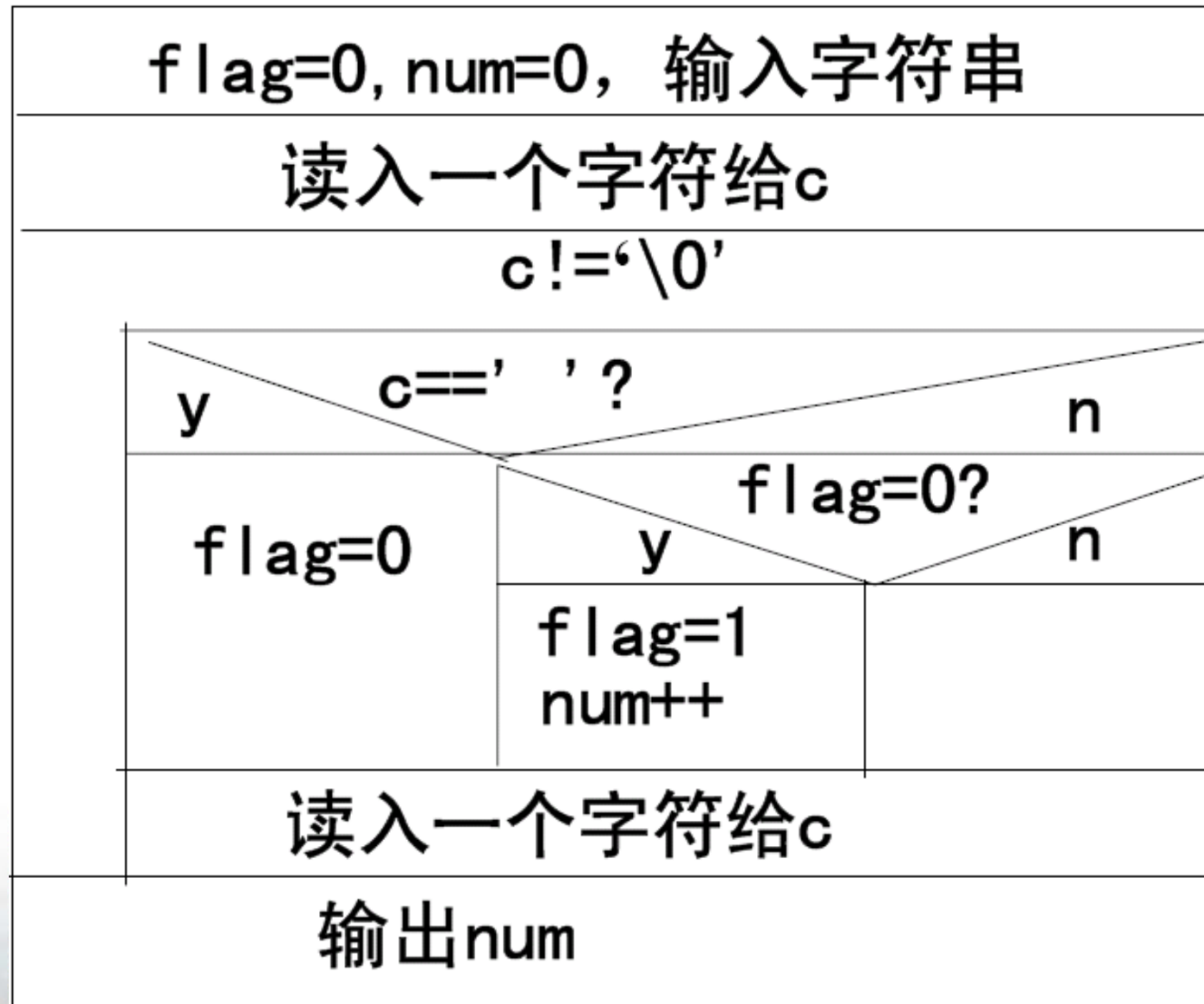
- ◆ 1. 读入一整句话;
- ◆ 2. 用什么区分单词? 空格后面的字母
- ◆ 3. 每次读入一个字符时, 如果是字母, 就要判断前一个是不是空格, 是则新单词开始, 不是则继续
  - 需要记住前一个字符的状态:



北京大学



## 字符数组应用举例(3)





## 字符数组应用举例(3)

```
int main()
{ char str[80];
  int i, num = 0, flag = 0;
  char c;
  cin.getline(str, 80);
  for(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;
}
```



北京大学



好好想想,有没有问题?

谢谢!



北京大学