

# 利用数组处理批量 数据

主讲人：陈莹老师

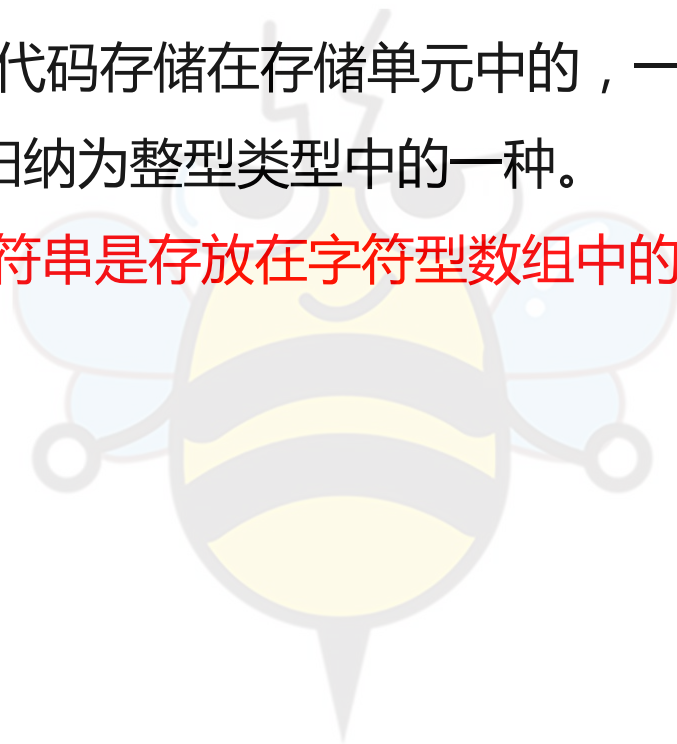


哎上课

哎上课 升本科

## 6.3 字符数组

- 字符型数据是以字符的 ASCII 代码存储在存储单元中的，一般占一个字节。
- 在 C 99 标准中，把字符类型归纳为整型类型中的一种。
- C 语言中没有字符串类型，字符串是存放在字符型数组中的。



哎上课 升本课

## 6.3.1 怎样定义字符数组

字符数组：存放字符数据的数组。

一维字符数组：存放一个字符串（每个数组元素存放一个字符）

二维字符数组：存放多个字符串（行数是字符串的个数）

### ★ 字符数组的定义

❖ 形式：

- char 数组名[常量表达式]
- char 数组名[常量表达式][常量表达式]
- 常量表达式：整数、字符、符号常量

例 `char c[10], ch[3][4];`

## 6.3.1 怎样定义字符数组

```
char c [ 10 ] ;  
c [ 0 ] = ' I ' ; c [ 1 ] = ' ' ; c [ 2 ] = ' a ' ; c [ 3 ] = ' m ' ; c [ 4 ] = ' ' ;  
c [ 5 ] = ' h ' ; c [ 6 ] = ' a ' ; c [ 7 ] = ' p ' ; c [ 8 ] = ' p ' ; c [ 9 ] = ' y ' ;
```

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
I		a	m		h	a	p	p	y

❖ 可以用整型数组存放字符型数据，但浪费存储空间。

如：

```
int c [ 10 ] ;
```

```
c [ 0 ] = ' a ' ; //合法，但浪费存储空间
```

**在计算机中int（32位系统中）占4个字节，Char占1个字节**

## 6.3.2 字符数组的初始化

- ❖ 逐个字符赋值
- ❖ 用字符串常量

① 完全初始化：

例 `char ch[5]={'H','e','l','l','o'};`

H	e	l	l	o
ch[0]	ch[1]	ch[2]	ch[3]	ch[4]

例 `char ch[ ]={'H','e','l','l','o'};`

H	e	l	l	o
ch[0]	ch[1]	ch[2]	ch[3]	ch[4]

## 6.3.2 字符数组的初始化

### ② 不完全初始化

例 `char ch[5]={'B','o','y'};`

B	o	y	\0	\0
ch[0]	ch[1]	ch[2]	ch[3]	ch[4]

### ③ 字符个数大于数组长度

例 `char ch[4]={'H','e','l','l','o'};`

H	e	l	l	o
ch[0]	ch[1]	ch[2]	ch[3]	ch[4]

有问题!

## 6.3.2 字符数组的初始化

### ④ 定义和初始化二维数组

例 `char diamond[][5]={{' ',' ','*'},{' ','*',' ','*'},  
{'*',' ',' ',' ','*'},{' ','*',' ','*'},{' ',' ','*'}};`

`diamond[0]`

		*	\0	\0
--	--	---	----	----

`diamond[1]`

	*		*	\0
--	---	--	---	----

`diamond[2]`

*				*
---	--	--	--	---

`diamond[3]`

	*		*	\0
--	---	--	---	----

`diamond[4]`

		*	\0	\0
--	--	---	----	----

## 6.3.3 怎样引用字符数组中的元素

### 例6.6 输出一个字符串

```
#include <stdio.h>
void main()
{ char c[10]={'I',' ','a','m',' ','a',' ','b','o','y'};
  int i;
  for(i=0;i<10;i++)
    printf("%c",c[i]);
  printf("\n");
}
```

0	I
1	
2	a
3	m
4	
5	a
6	
7	b
8	o
9	y

## 6.3.3 怎样引用字符数组中的元素

### 例6.7 输出一个钻石图形

```
#include <stdio.h>
void main()
{ char diamond[ ][5]={{' ',' ','*'},{' ','*',' ','*'},{'*',' ',' ',' ','*'},
                      {' ','*',' ','*'},{' ',' ','*'}};

  int i,j;
  for(i=0;i<5;i++)
    {for(j=0;j<5;j++)
      printf("%c",diamond[i][j]);
      printf("\n");
    }
}
```

运行结果:

```
  *
 * *
*   *
*   *
 *
```

## 6.3.4 字符串和字符串结束标志

### ★ 字符串和字符串结束标志

❖ 字符串：用双引号括起的若干字符，如：“china”

可将其存放在一维或两维字符型数组中。

无字符串变量，用字符数组处理字符串

❖ 字符串结束标志：‘\0’，(既无动作,又不显示)

● 字符串的长度：第一个 ‘\0’ 以前字符的个数

● 在字符型数组或字符串中遇 ‘\0’，即认为该字符串结束。

● 系统对字符串常量自动加一个 ‘\0’ 作为结束符。

```
printf( "china" );
```

c	h	i	n	a	\0
---	---	---	---	---	----

## 6.3.4 字符串和字符串结束标志

❖ 用字符串常量初始化字符数组

```
例 char ch[6]={"Hello"};  
char ch[6]="Hello";  
char ch[]="Hello";  
char ch[]={'H','e','l','l','o','\0'};
```

H	e	l	l	o	\0
ch[0]	ch[1]	ch[2]	ch[3]	ch[4]	ch[5]

```
例 char ch[10]={"Hello"};
```

H	e	l	l	o	\0	\0	\0	\0	\0
ch[0]	ch[1]	ch[2]	ch[3]	ch[4]	ch[5]	ch[6]	ch[7]	ch[8]	ch[9]

## 6.3.4 字符串和字符串结束标志

❖ 用字符串常量初始化字符数组

### 二维字符数组初始化

例 `char fruit[][7]={"Apple","Orange",  
"Grape","Pear","Peach"};`

<code>fruit[0]</code>	A	p	p	l	e	\0	\0
<code>fruit[1]</code>	O	r	a	n	g	e	\0
<code>fruit[2]</code>	G	r	a	p	e	\0	\0
<code>fruit[3]</code>	P	e	a	r	\0	\0	\0
<code>fruit[4]</code>	P	e	a	c	h	\0	\0

## 6.3.4 字符串和字符串结束标志

说明：字符数组并不要求它的最后一个字符为 ' \0 '，甚至可以不包含 ' \0 '。像以下这样写完全是合法的：

```
char c [ 5 ] = { ' c ' , ' h ' , ' i ' , ' n ' , ' a ' } ;
```

由于系统在处理字符串常量存储时会自动加一个 ' \0 '，因此，为了使处理方法一致，便于测定字符串的实际长度，以及在程序中作相应的处理，在字符数组中也常常人为地加上一个 ' \0 '。

例如：

```
char c [ 6 ] = { ' c ' , ' h ' , ' i ' , ' n ' , ' a ' , ' \0 ' } ;
```

这样做，便于引用字符数组中的字符串。

哎上课 升本科

## 6.3.4 字符串和字符串结束标志

如定义了以下的字符数组：

```
char c [ ] = { "Cprogram." } ;
```

由于系统自动在最后一个字符后面加了一个 ' \0 ' ，因此 c 数组的存储情况如下：

C		p	r	o	g	r	a	m	.	\0
---	--	---	---	---	---	---	---	---	---	----

若从键盘输入 "Hello" 分别赋给 c 数组中前面 5 个元素。不加 ' \0 ' 的话，字符数组的字符如下：

H	e	l	l	o	g	r	a	m	.	\0
---	---	---	---	---	---	---	---	---	---	----

如果想输出字符数组中的字符串，则会连续输出：

Hellogram.

如果在 "Hello" 后面加一个 ' \0 ' ，它取代了第 6 个字符 "g" 。在数组中的存储情况为

H	e	l	l	o	\0	r	a	m	.	\0
---	---	---	---	---	----	---	---	---	---	----

```
printf ( "%s\n" , c ) ; // 输出数组 c 中的字符串
```

输出结果为：**Hello**

## 6.3.5 字符数组的输入输出

❖ 逐个字符I/O : %c

例 用%c

```
void main()
{ char str[5];
  int i;
  for(i=0;i<5;i++)
    scanf("%c", &str[i]);
  for(i=0;i<5;i++)
    printf("%c", str[i]);
}
```

输入: China ↵

输出: China

输入: Program ↵

输出: Progr

## 6.3.5 字符数组的输入输出

❖ 整个字符串I/O : %s

用字符数组名, 不要加&, 在 C 语言中数组名代表该数组的起始地址

输入串长度 < 数组维数

遇空格或回车结束

自动加 '\0'

例 用%s

```
void main()
```

```
{ char str[5];
```

```
scanf("%s", str);
```

```
printf("%s", str);
```

```
}
```

输入: China ↵

输出: China

输入: Program ↵

输出: Progr

用字符数组名,  
遇 '\0' 结束

## 6.3.5 字符数组的输入输出

❖注意：

(1) 输出的字符中不包括结束符 ' \0 ' 。

(2) 用 "%s" 格式符输出字符串时，printf 函数中的输出项是字符数组名，而不是数组元素名。写成下面这样是不对的：

```
printf ( "%s" , c [ ] ) ;
```

(3) 如果数组长度大于字符串的实际长度，也只输出到遇 ' \0 ' 结束。

(4) 如果一个字符数组中包含一个以上 ' \0 ' ，则遇第一个 ' \0 ' 时输出就结束。

```
#include <stdio.h>
void main()
{
    char a[ ]={'h','e','l','\0','l','o','\0'};
    printf("%s",a);
}
```

h	e	l	\0	l	o	\0
---	---	---	----	---	---	----

输出：hel

## 6.3.5 字符数组的输入输出

(5) 如果利用一个 scanf 函数输入多个字符串，则应在输入时以空格分隔；

scanf中%s输入时,遇空格或回车结束

```
#include <stdio.h>
void main()
{ char a[15],b[5],c[5];
  scanf("%s%s%s",a,b,c);
  printf("a=%s\nb=%s\nc=%s\n",a,b,c);
  scanf("%s",a);
  printf("a=%s\n",a);
}
```

运行情况:

输入: How are you?

输出: a=How

b=are

c=you?

输入: How are you?

输出: a=How

H	o	w	\0																	
a	r	e	\0																	
y	o	u	?	\0																

## 6.3.5 字符数组的输入输出

(6) scanf 函数中的输入项如果是字符数组名，不要再加地址符 &，因为在 C 语言中数组名代表该数组的起始地址

```
scanf ( "%s" , &.str ) ; ( × ) //str 前面不应加 &
```

若数组名为 c，占 6 个字节。数组名 c 代表地址 2000。可以用下面的输出语句得到数组的起始地址。

	c数组
2000	C
2001	h
2001	i
2003	n
2004	a
2005	\0

```
printf ( "%o" , c ) ;
```

// 用八进制形式输出数组 c 的起始地址

输出数组 c 的起始地址 ( 例如 2000 )。可知数组名 c 代表数组起始地址。

```
printf ( "%s" , c ) ;
```

//按字符数组名 c 找到其数组起始地址，然后逐个输出其中的字符，直到遇 ' \0 为止。

# 随堂测

1. 以下语句中，s 不能作为字符串使用的是（ ）。

A.char s[]=" shanghai" ;

B.char s[]={ "shanghai" };

C.char s[9]={ 's' , 'h' , 'a' , 'n' , 'g' , 'h' , 'a' , 'i' };

D.char s[8]= { 's' , 'h' , 'a' , 'n' , 'g' , 'h' , 'a' , 'i' };

2. 设有定义：

```
1 char a[]="China";
```

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

则以下叙述正确的是（ ）。

A. 数组 a 和数组 b 等价

B. 数组 a 和数组 b 的长度相同

C. 数组 a 的长度大于数组 b 的长度

D. 数组 a 的长度小于数组 b 的长度

哎上课 升本科

# 随堂测

3. 下面程序的运行结果是 ( )。

```
# include <stdio.h>
void main()
{
char a[]="abcdef";
a[4]='\0';
printf("%s\n",a);}
```

A.abcd

B.bcde

C.cdef

D. 不确定

4. 以下程序的输出结果是 ( )。

```
# include <stdio.h>
# include <string.h>
void main()
{
char s[][5]={"ABCD","EFGH","IJKL","MNOP"};
int i;
for(i=1;i<3;i++)
printf("%s",s[i]); }
```

A.ABCDEFGH

B.EFGHIJKL

C.IJKLMNOP

D.MNOP



哎上课 升本科

## 6.3.6 使用字符串处理函数

➤ 在 C 函数库中提供了一些用来专门处理字符串的函数，包含在头文件 `string.h` 中

### ❖ 1. 字符串输出函数：`puts`

- 格式：`puts` ( 字符数组 )
- 功能：向显示器输出一个字符串（输出完，换行）
- 说明：字符数组必须以 `'\0'` 结束。可以包含转义字符。  
输出时 `'\0'` 转换成 `'\n'`，即输出字符后换行。

这里是将  
`'\0' → '\n'`  
因此光标移  
到下行

运行结果：  
china  
beijing  
china  
WUHAN

例：

```
#include <stdio.h>
void main( )
{ char a1[ ]="china\nbeijing" ;
  char a2[ ]="china\0beijing" ;
  puts(a1); puts(a2);
  puts("WUHAN" );
}
```

## 6.3.6 使用字符串处理函数

### ❖ 2. 字符串输入函数 `gets`

- 格式：`gets` (字符数组)
- 功能：从键盘输入一个以回车结束的字符串放入字符数组中，并自动加 `'\0'`，得到一个函数值，该函数值是数组的起始地址。
- 说明：输入串长度应小于字符数组维数

#### 注意：

`puts`和`gets`函数只能输入输出一个字符串

#### 错

```
puts(str1,str2)
```

```
gets(str1,str2)
```

哎上课 升本科

## 6.3.6 使用字符串处理函数

例：gets和scanf输入比较

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

```
void main( )
```

```
{ char a1[15], a2[15];
```

```
  gets(a1);
```

```
  scanf(“%s”,a2);
```

```
  printf(“a1=%s\ n”,a1);
```

```
  printf(“a2=%s\ n”,a2);
```

```
}
```

在scanf 中遇空格字符串便结束了，而gets 中，却将空格作为字符存入字符型数组中。

输入： china beijing ↵

china beijing ↵

输出： a1=china beijing

a2=china

哎上课 升本科

## 6.3.6 使用字符串处理函数

### ❖ 3. 字符串连接函数 `strcat`

- 格式：`strcat` (字符数组1,字符数组2)
- 功能：把字符数组2连到字符数组1后面
- 返回值：返回字符数组1的首地址
- 说明：①字符数组1必须足够大  
②连接前,两串均以 '\0' 结束;连接后,串1的 '\0' 取消，只在新串最后保留 '\0'。

例：

```
#include <stdio.h>
void main( )
{ char str1[30]={“People’s Republic of “};
  char str2[]={“China”};
  printf (“%s\n”,strcat(str1,str2));
}
```

```
str1: People’s Republic of \0
str2: china\0
str1: People’s Republic of China\0
```

## 6.3.6 使用字符串处理函数

### ❖ 4. 字符串拷贝函数 `strcpy`

- 格式：`strcpy`(字符数组1,字符串2)
- 功能：将字符串2，拷贝到字符数组1中去
- 返回值：返回字符数组1的首地址

如：

```
char str1 [10] , str2 [] = " China";  
strcpy (str1, str2) ;
```

执行后，str1状态如下：

C	h	i	n	a	\0	\0	\0	\0	\0
---	---	---	---	---	----	----	----	----	----

## 6.3.6 使用字符串处理函数

● 说明：

- ① 字符数组1必须足够大，长度不应小于字符串2的长度
- ② 符数组1必须是数组名形式 ( str1 ) ,字符串2可以是字符数组名，也可以是字符串常量

如：

```
strcpy (str1, " China");
```

- ③ 拷贝时 '\0' 一同拷贝，取代字符数组的前取代字符数组 1 中的前面 6 个字符，最后 4 个字符并不一定是 ' \0 '，而是 str1 中原有的最后 4 个字节的内容。
- ④ 不能使用赋值语句为一个字符数组赋值

例 char str1[20],str2[20];

str1={"Hello!"}; (×) // 企图用赋值语句将一个字符串常量直接赋给一个字符数组

str2=str1; (×) // 企图用赋值语句将一个字符数组直接赋给另一个字符数组

用赋值语句只能将一个字符 赋给一个字符型变量或字符数组元素。

## 6.3.6 使用字符串处理函数

● 说明：

⑤ 可以用 `strncpy` 函数将字符串 2 中前面个字符复制到字符数组 1 中去。

例如：`strncpy ( str1 , str2 , 2 )`；作用是将 `str2` 中最前面 2 个字符复制到 `str1` 中，取代 `str1` 中原有的最前面 2 个字符。但复制的字符个数 `n` 不应多于 `str1` 中原有的字符（不包括 `'\0'`）

哎上课 升本科

## 6.3.6 使用字符串处理函数

### 例 strcpy与strcat应用举例

```
#include <stdio.h>
void main()
{ char destination[25];
  char blank[] = " ", c[] = "C++",
    turbo[] = "Turbo";
  strcpy(destination, turbo);
  strcat(destination, blank);
  strcat(destination, c);
  printf("%s\n", destination);
}
```

**Turbo C++**

0	T
1	u
2	r
3	b
4	o
5	\0
6	
7	
8	
9	
24	

0	T
1	u
2	r
3	b
4	o
5	
6	\0
7	
8	
9	
24	

0	T
1	u
2	r
3	b
4	o
5	
6	C
7	+
8	+
9	\0
24	

## 6.3.6 使用字符串处理函数

### ❖ 5. 字符串比较函数 `strcmp`

- 格式：`strcmp`(字符串1,字符串2)

如：

```
strcmp ( str1 , str2 ) ;  
strcmp ( "China" , " Korea" ) ;  
strcmp ( str1 , " Beijing" ) ;
```

- 功能：比较两个字符串
- 比较规则：对两串从左向右逐个字符比较（ASCII码），直到遇到不同字符或 '\0' 为止。

(1) 如全部字符相同，则认为两个字符串相等；

(2) 若出现不相同的字符，则以第 1 对不相同的字符的比较结果为准

"A" < " B" , " a" > " A" , " computer" > " compare" , " these" > " that" ,  
" 1A" > " \$20" , " CHINA" > "CANADA" , " DOG" < " cat"  
" Tsinghua" > " TSINGHUA"

## 6.3.6 使用字符串处理函数

- 返回值：返回int型整数函数值。其值是ASCII码的差值
  - a. 若字符串1 < 字符串2，返回**负整数**
  - b. 若字符串1 > 字符串2，返回**正整数**
  - c. 若字符串1 == 字符串2，返回**零**
- 说明：**字符串比较不能用 “!=、==、>、>=、<、<=、”** 等关系运算符！,必须用strcmp，虽然编译无错，但结果不对

错

```
if(str1==str2) printf(“yes”);  
if (str1>str2)  
printf ( “yes” ) ;
```

对

```
if(strcmp(str1,str2)==0) printf(“yes”);  
if (strcmp (str1, str2) >0)  
printf ( “yes” ) ;
```

## 6.3.6 使用字符串处理函数

例：字符比较

```
#include <stdio.h>
void main( )
{ int i,j,k;
  char a1[ ]="wuhan", a2[ ]="beijing";
  i=strcmp(a1,a2);
  j=strcmp("china", "korea");
  k=strcmp(a2, "beijing" );
  printf("i=%d\ nj=%d\ nk=%d\ n",i,j,k);
}
```

运行结果：

```
i=21    i=w-b=119-98=21
j=-8    j=c-k=99-107=-8
k=0
```

## 6.3.6 使用字符串处理函数

### ❖ 6. 字符串长度函数 `strlen`

- 格式：`strlen`(字符数组)
- 功能：计算字符串长度

如：

```
char str [ 10 ] = " China" ;  
printf ( "%d" , strlen ( str ) )
```

结果为：5

- 返回值：返回字符串实际长度，**不包括 '\0' 在内**

例 对于以下字符串，`strlen(s)`的值为：

(1) `char s[10]={‘A’, ‘\0’, ‘B’, ‘C’, ‘\0’, ‘D’};`

(2) `char s[ ]=“\t\v\\\0will\n”;`

(3) `char s[ ]=“\x69\082\n”;`

答案：1 3 1

## 6.3.6 使用字符串处理函数

### ❖ 6. 字符串长度函数 `strlen`

- 格式：`strlen`(字符数组)
- 功能：计算字符串长度

如：

```
char str [ 10 ] = " China" ;  
printf ( "%d" , strlen ( str ) )
```

结果为：5

- 返回值：返回字符串实际长度，**不包括 '\0' 在内**

例 对于以下字符串，`strlen(s)`的值为：

(1) `char s[10]={‘A’, ‘\0’, ‘B’, ‘C’, ‘\0’, ‘D’};`

(2) `char s[ ]=“\t\v\\\0will\n”;`

(3) `char s[ ]=“\x69\082\n”;`

答案：1 3 1

## 6.3.6 使用字符串处理函数

例：测试字符串长度

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

```
void main( )
```

```
{ char a1[10]=" china" ;
```

```
  printf ("%d\ n",strlen(a1));
```

```
  printf ("%d\ n", strlen("beijing\ 0wuhan"));
```

```
}
```

运行结果： 5  
7

哎上课 升本科

## 6.3.6 使用字符串处理函数

❖ 7. 大写字母转换成小写字母函数 `strlwr`

● 格式：`strlwr`(字符串)

❖ 8. 小写字母转换成大写字母函数 `strupr`

● 格式：`strupr`(字符串)

例：字符转换

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

```
void main( )
```

```
{ char a1[6]="CHinA", a2[ ]="wuHAn" ;
```

```
    printf ("%s\ n",strlwr(a1)); printf ("%s\ n",strupr(a2));
```

```
}
```

运行结果：  
china  
WUHAN



# 随堂测

5. 当执行下面程序时，如果输入 abc, 则输出结果是 ( )。

```
# include <stdio.h>
# include <string.h>
void main()
{ char str[10]="12345";
  gets(str);
  strcat(str,"def");
  printf("%s\n", str);
}
```

A.abcdef

B.12345def

C.12345abc

D.abc45def

6. 以下程序的运行结果是 ( )。

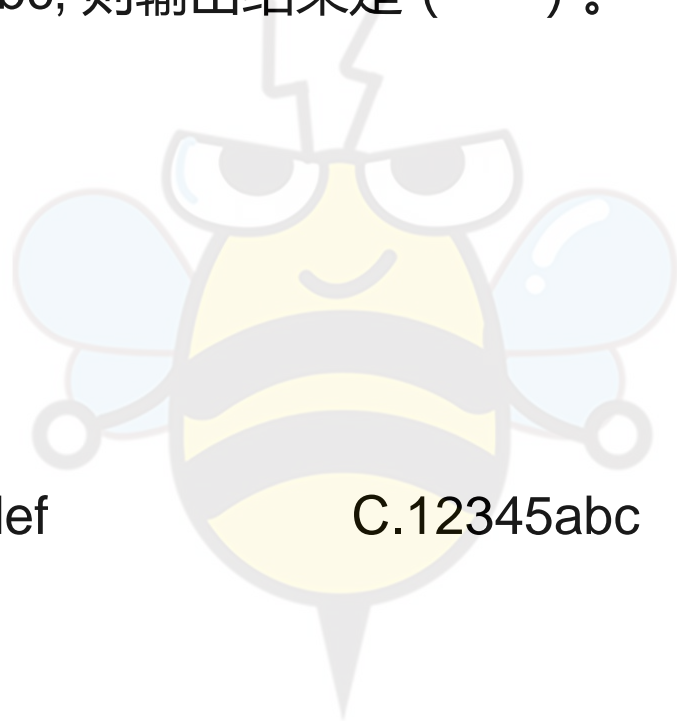
```
# include <stdio.h>
void main()
{
  char a[10]="abcd";
  printf("%d,%d\n", strlen(a), sizeof(a));
}
```

A.4 , 7

B.4 , 8

C.8 , 8

D.4 , 10



哎上课 升本科

# 加油，升本路上的孤勇者！

关注哎上课  
更多资料  
等你来拿



哎上课 升本科