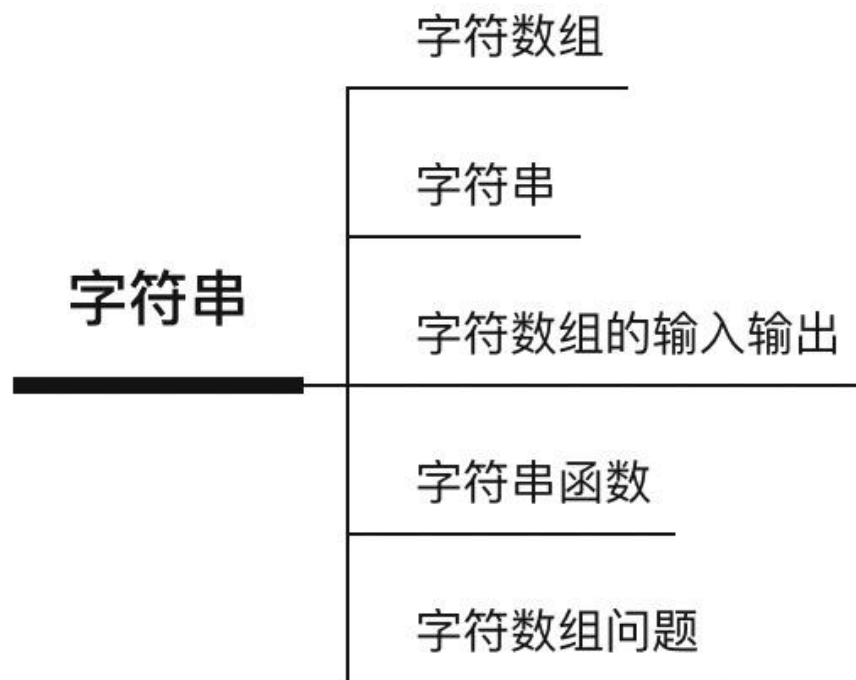


# C语言基础

## 精讲1



# 真题演练 - 4



## ▶ 4.1 字符数组

字符数组定义格式：

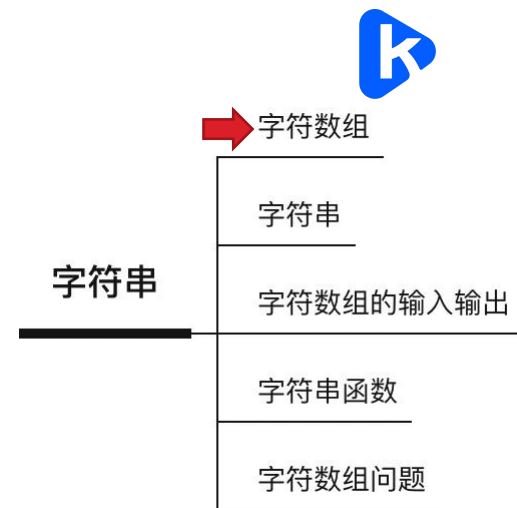
类型说明符 数组名[常量表达式];

例如： `char name[20];` //定义字符数组

字符数组引用格式：

数组名[下标]      //下标只能为整型常量或整型表达式

**注意：字符数组可整体引用**



## 4.1 字符数组



→ 字符数组

字符串

字符串

字符数组的输入输出

字符串函数

字符数组问题

### 字符数组初始化

- 1) 逐个字符赋给数组中各元素 `char a[5]={'a','b','c','d','e'};`
- 2) 利用字符串给字符数组赋初值 `char a[10] = { "hello" };`
- 3) 省略数组长度 `char a[] = "hello";`

注意：字符数组长度  $\geq$  字符个数 + 1;

char a[5]={'a','b','c','d','e'};									
a	b	c	d	e					
a[0]	a[1]	a[2]	a[3]	a[4]					
char a[10]="C语言学习";									
C	语	言	学	习	\0				
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
char a[]="C语言学习";									
C	语	言	学	习	\0				
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

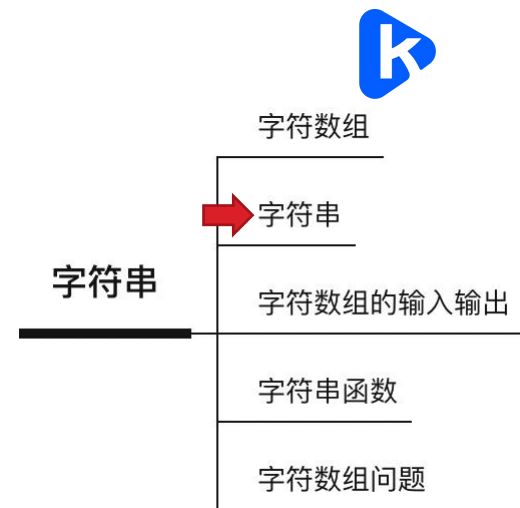
## 4.2 字符串

### 1) 字符串定义

字符串是字符的集合，C语言中没有字符串数据类型，因为字符串以常量形式出现或存储于数组中。

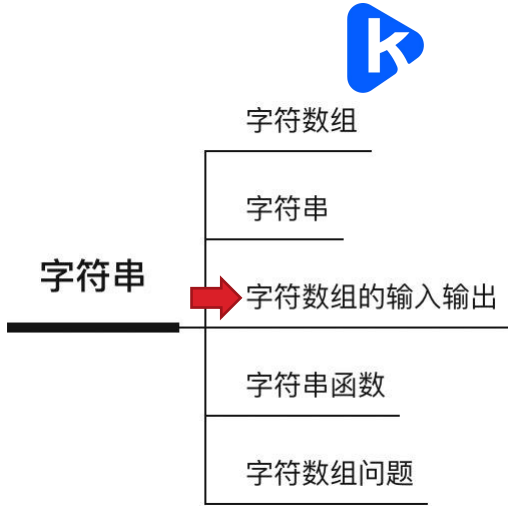
### 2) 字符串标志

字符串最后以空字符' \0 ' 作为终止符，一个字符串的长度为字符的数量，不包括终止符。



## ▶ 4.3 字符数组的输入输出

	输入	输出
格式化单	<code>scanf(“%c”,&amp;变量名)</code>	<code>printf(“%c”,变量名)</code>
格式化多	<code>scanf(“%s”,数组名)</code>	<code>printf(“%s”,数组名)</code>
单字符	<code>变量名 = getchar()</code>	<code>putchar(变量名)</code>
字符串	<code>gets(数组名)</code>	<code>puts(数组名)</code>

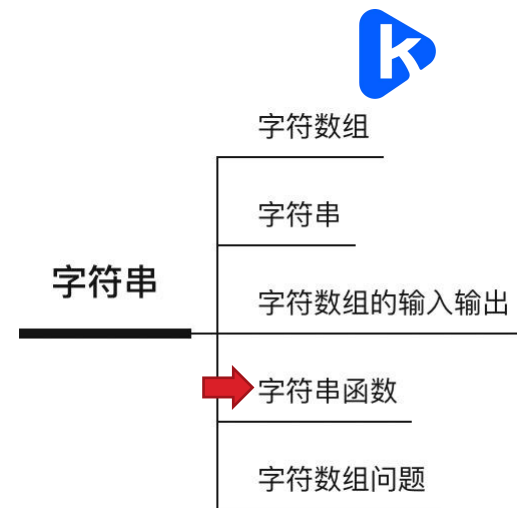


注意：

- 1) 字符数组以 ‘\0’ 结束，可整体引用
- 2) 格式化输入字符串时不能接收空格，字符串输入函数`gets()`可接收空格，或通过`%[^\\n]`实现

## 4.4 字符串函数

- |            |  |
|------------|--|
| 1) 字符串连接函数 | strcat ( 字符数组, 字符串 )                     |
| 2) 字符串复制函数 | strcpy ( 字符数组, 字符串 )                     |
| 3) 字符串比较函数 | strcmp ( 字符数组, 字符串 ) //大于 1; 小于 -1; 等于 0 |
| 4) 字符串长度函数 | strlen ( 字符常量或字符数组 )                     |





## ▶ 字符数组问题

广东工业大学 2019年

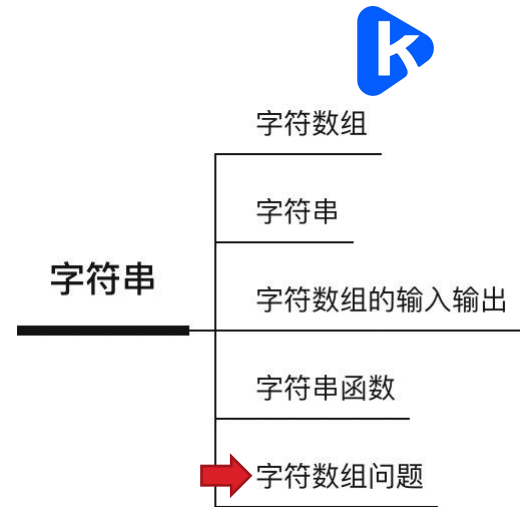
12.以下能正确定义一维数组的选项是\_\_\_\_\_.

```
A int a[5]={0,1,2,3,4,5};
```

```
B char a[ ]={0,1,2,3,4,5};
```

```
C char a={'a','b','c'};
```

```
D int a[5]="0123";
```



## ► 字符数组问题

广东工业大学 2019年

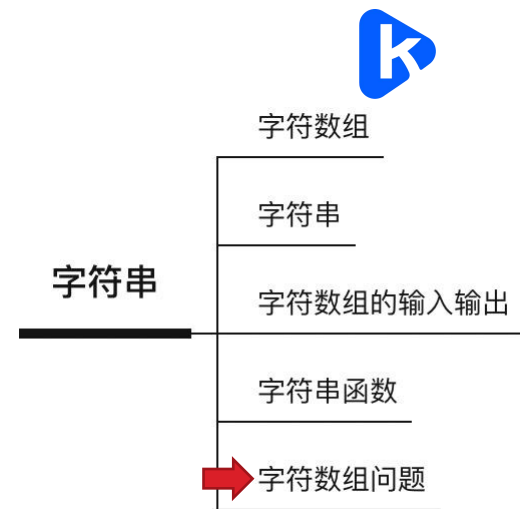
12. 以下能正确定义一维数组的选项是 **B**.

A `int a[5]={0,1,2,3,4,5};`

B `char a[ ]={0,1,2,3,4,5};`

C `char a={'a','b','c'};`

D `int a[5]="0123";`



## ▶ 字符数组问题

北京航空航天大学 2018年

8.对于以下C程序，其正确的输出是 ( )

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

```
int main(){
```

```
    char str1[ ]="Hello";
```

```
    char str2[ ]="Hello";
```

```
    if (str1 == str2) printf("Equal\n");
```

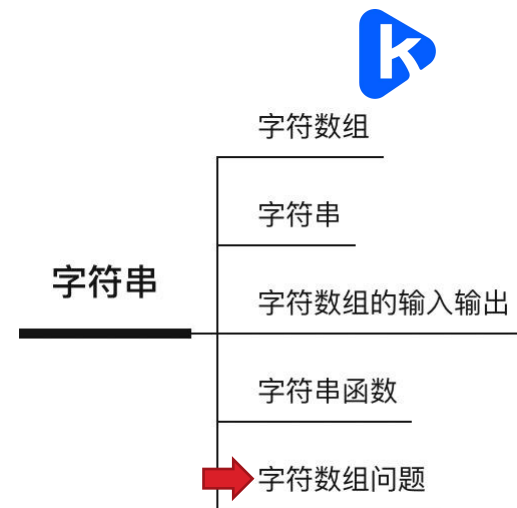
```
    else printf("Unequal\n");
```

```
    return 0;
```

```
}
```

A.Unequal;      B.Equal;      C.该程序无法通过编译;

D.该程序运行时出错。



## ► 字符数组问题

北京航空航天大学 2018年

8.对于以下C程序，其正确的输出是 ( A )

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

```
int main(){
```

```
    char str1[ ]="Hello";
```

```
    char str2[ ]="Hello";
```

```
    if (str1 == str2) printf("Equal\n");
```

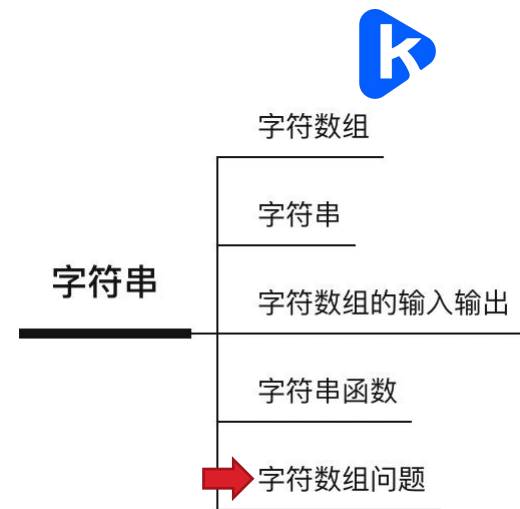
```
    else printf("Unequal\n");
```

```
    return 0;
```

```
}
```

A.Unequal;      B.Equal;      C.该程序无法通过编译;

D.该程序运行时出错。



## ► 字符数组问题

北京航空航天大学 2019年

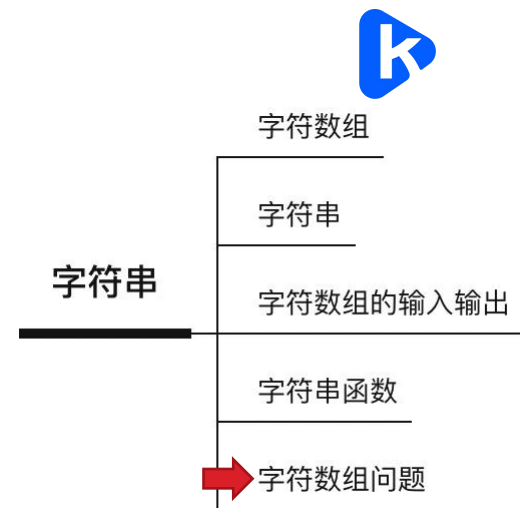
5.要将"Hello World"这个中间带空格的字符串正确地读入到一维数组 str 中, 应该使用的语句是\_\_\_\_\_。

A scanf("%s", &str);;

B scanf("%s", str);

C gets(&str);

D gets(str);。



## ► 字符数组问题

北京航空航天大学 2019年

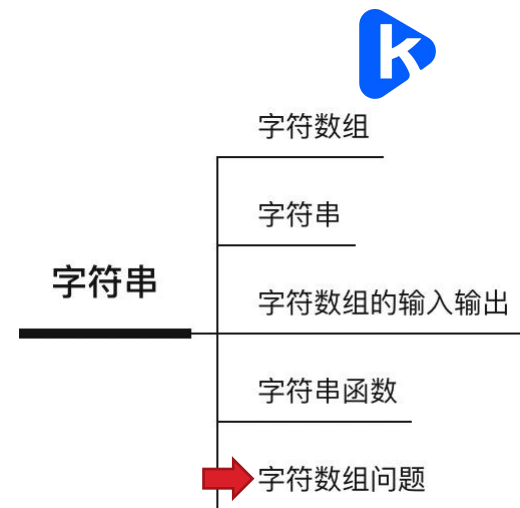
5.要将"Hello World"这个中间带空格的字符串正确地读入到一维数组 str 中, 应该使用的语句是\_\_D\_\_。

A scanf("%s", &str);;

B scanf("%s", str);

C gets(&str);

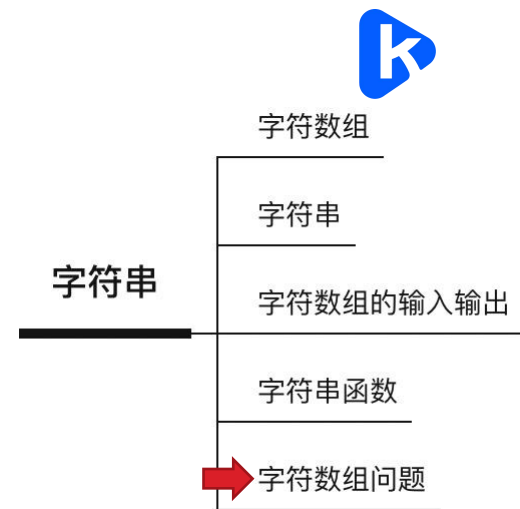
D gets(str);。



## ► 字符数组问题

7. 下列关于 C 语言数组的叙述中，错误的是\_\_\_\_\_。

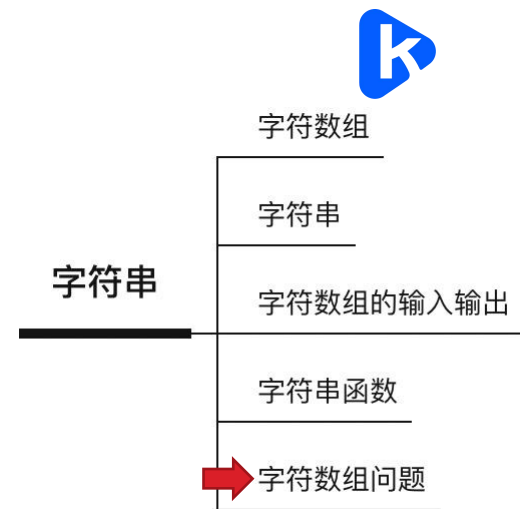
- A. 数组名代表了该数组的首地址;
- B. 数组变量必须占用连续的内存空间;
- C. 声明数组变量时就分配了相应的存储空间;
- D. 数组的大小存储在数组的第一个元素前面。



## ▶ 字符数组问题

7. 下列关于 C 语言数组的叙述中，错误的是\_\_D\_\_。

- A. 数组名代表了该数组的首地址;
- B. 数组变量必须占用连续的内存空间;
- C. 声明数组变量时就分配了相应的存储空间;
- D. 数组的大小存储在数组的第一个元素前面。



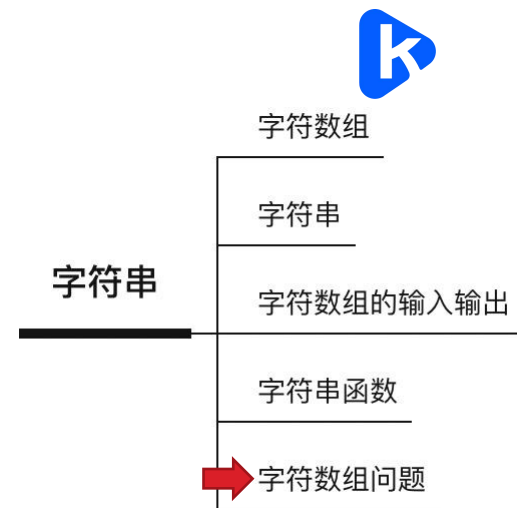


## ► 字符数组问题

北京航空航天大学 2021年

6. 下列函数 FUNC6 的功能是删除字符串 s 中的所有空白字符，包括 Tab 字符、回车符以及换行符。请在函数的空白处(符号-----处)填入合适内容，使函数完整。

```
FUNC6(char *s){  
    int i, t;  
    char c[80];  
    for(i=0,t =0; s[i]; i++)  
        if(!isspace(-----))  
            c[-----] =s[i];  
    c[t]= '\0' ;  
    strcpy(s, c);  
}
```

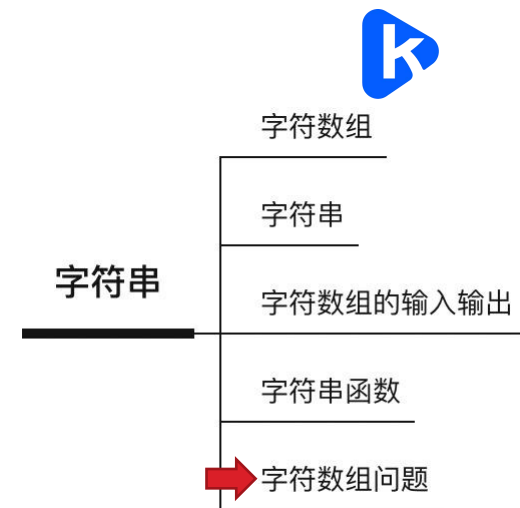


## ► 字符数组问题

北京航空航天大学 2021年

6. 下列函数 FUNC6 的功能是删除字符串 s 中的所有空白字符，包括 Tab 字符、回车符以及换行符。请在函数的空白处(符号-----处)填入合适内容，使函数完整。

```
FUNC6(char *s){  
    int i, t;  
    char c[80];  
    for(i=0,t=0; s[i]; i++)  
        if(!isspace( s[i] ))  
            c[ t++ ] =s[i];  
    c[t]= '\0' ;  
    strcpy(s, c);  
}
```

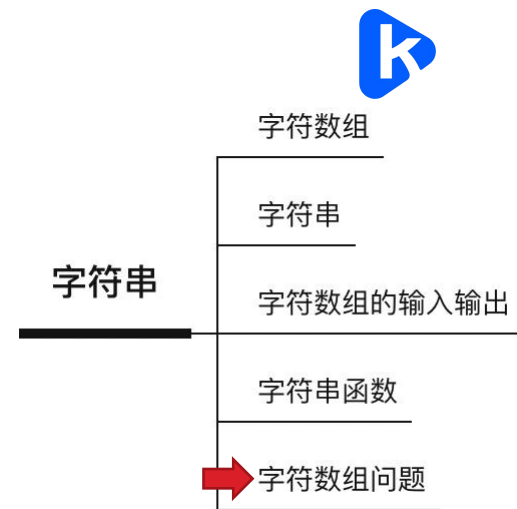


## ► 字符数组问题

北京航空航天大学 2019年

2. 下列函数 trim 的功能是删除字符串尾部的空白字符（包括空格符、制表符与换行符），并返回删除空白字符后剩余字符串的长度。请在函数的空白处（横线上方）填入合适的内容，使函数正确、完整。

```
int trim(chars[ ]){  
    int n;  
    for(n=strlen(s)-1; n≥0; n--)  
        if ( (1) ) (2) ;  
    s[ (3) ] = '\0' ;  
    return n;  
}
```

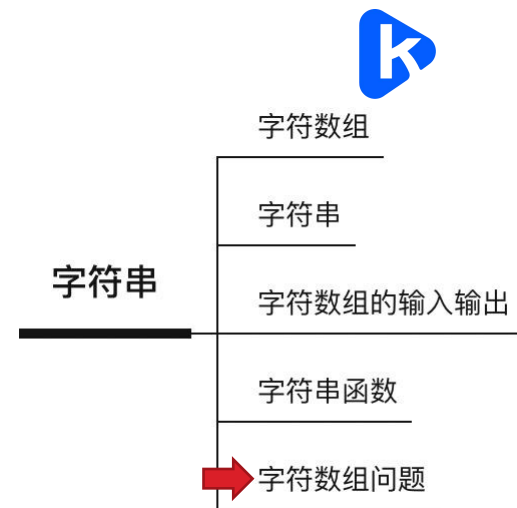


## ► 字符数组问题

北京航空航天大学 2019年

2. 下列函数 trim 的功能是删除字符串尾部的空白字符（包括空格符、制表符与换行符），并返回删除空白字符后剩余字符串的长度。请在函数的空白处（横线上方）填入合适的内容，使函数正确、完整。

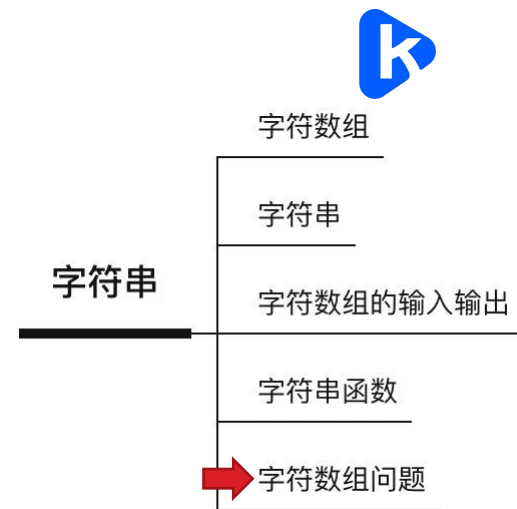
```
int trim(chars[ ]){  
    int n;  
    for(n=strlen(s)-1; n≥0; n--)  
        if ( s[n]!=' ' && s[n]!='\t' && s[n]!='\n' ) break ;  
    s[ ++n ] = '\0' ;  
    return n;  
}
```

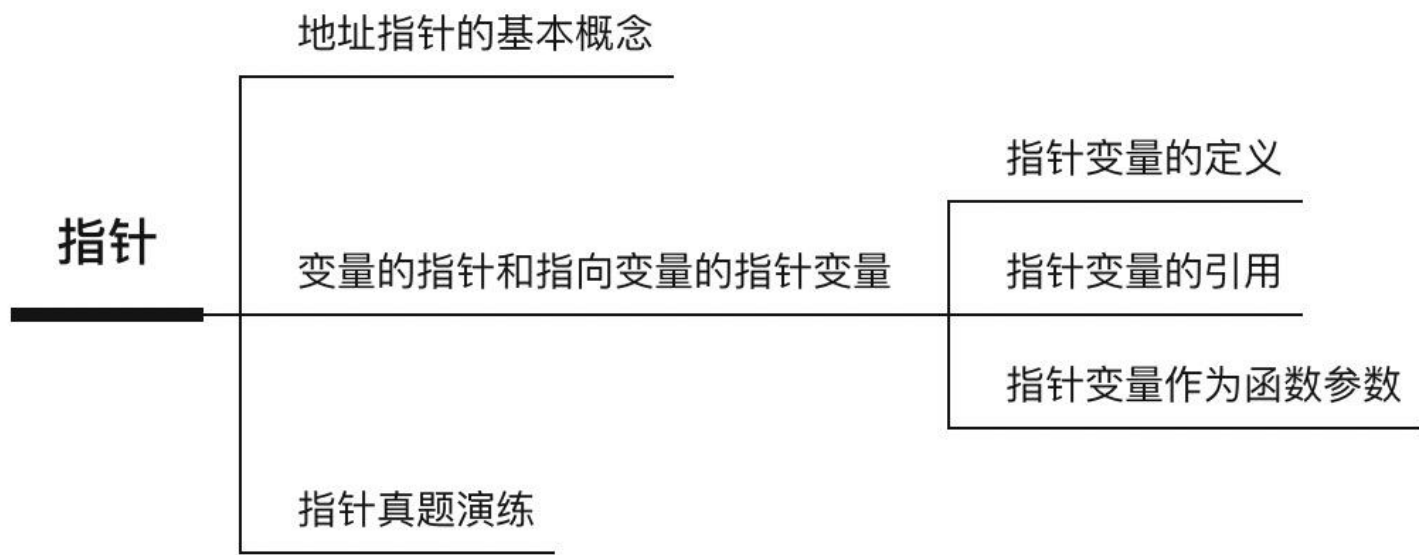


## ► 字符数组问题

北京师范大学 2012年

5 写一个函数，判断一字符串是否回文。回文是指顺序和倒读都一样的字符串。





## 5.1 地址指针的基本概念



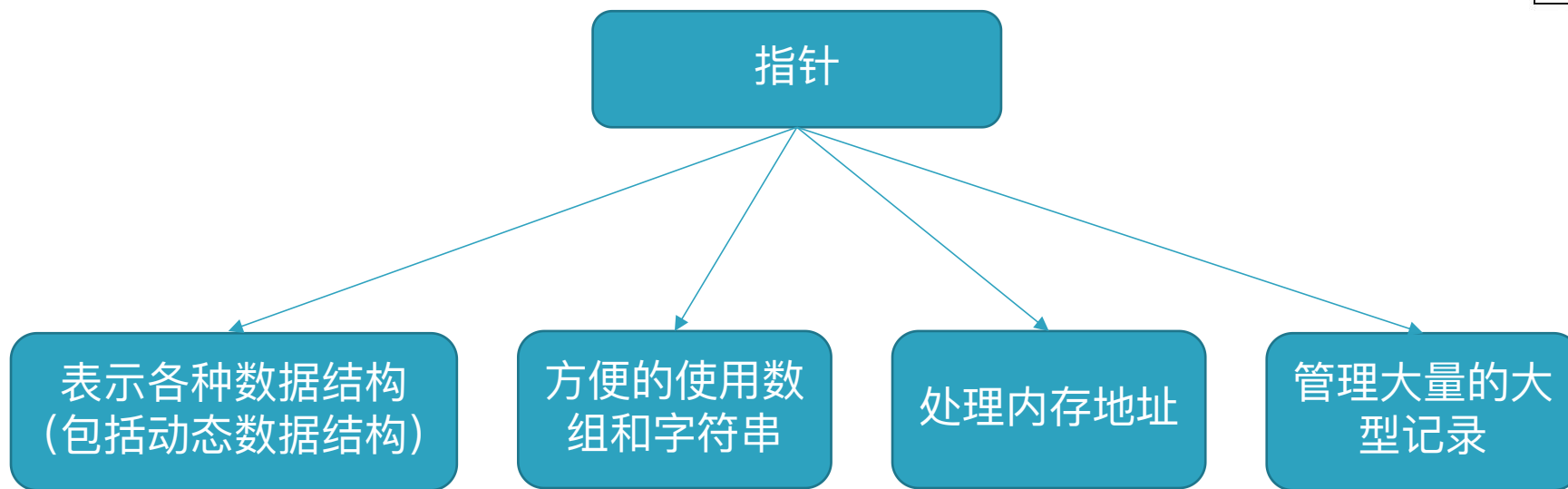
→ 地址指针的基本概念

指针

变量的指针和指向变量的指针变量

指针真题演练

3

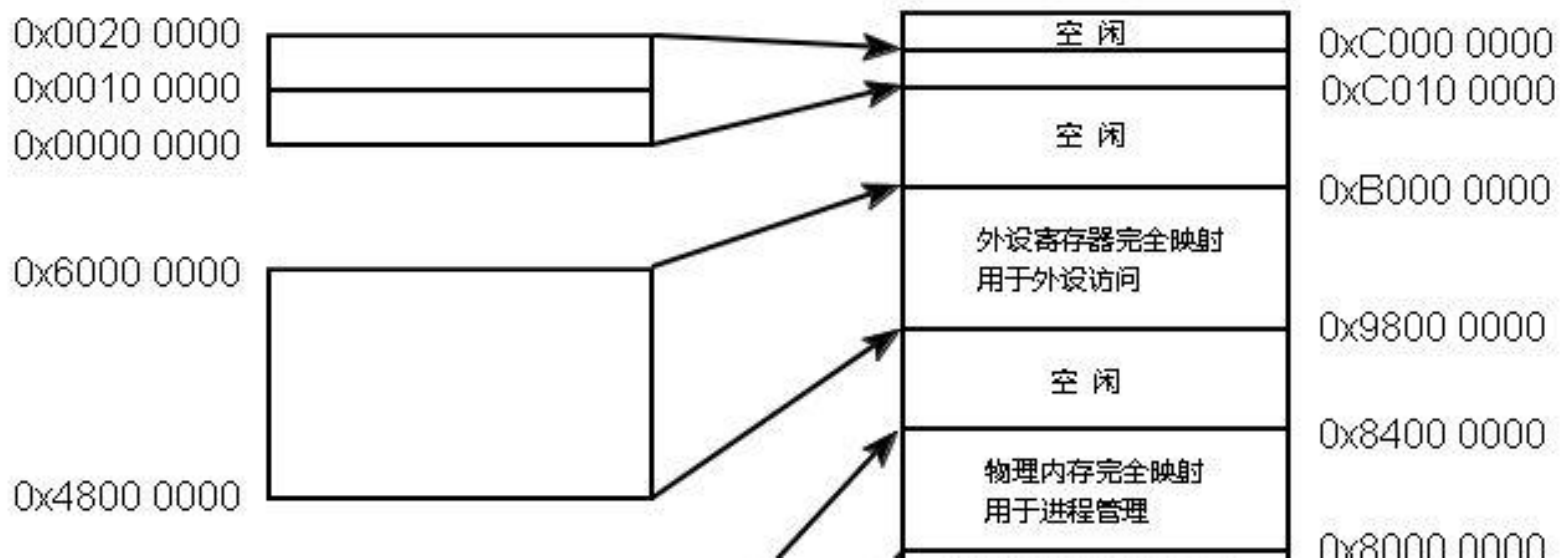


## 5.1 地址指针的基本概念

地址：内存单元的编号，根据地址找到该内存单元

指针变量：存储内存单元地址的变量。

数组或函数都是连续存放，通过访问指针变量取得数组或函数的首地址找到整个数组或函数，使程序精炼，高效。



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

指针真题演练

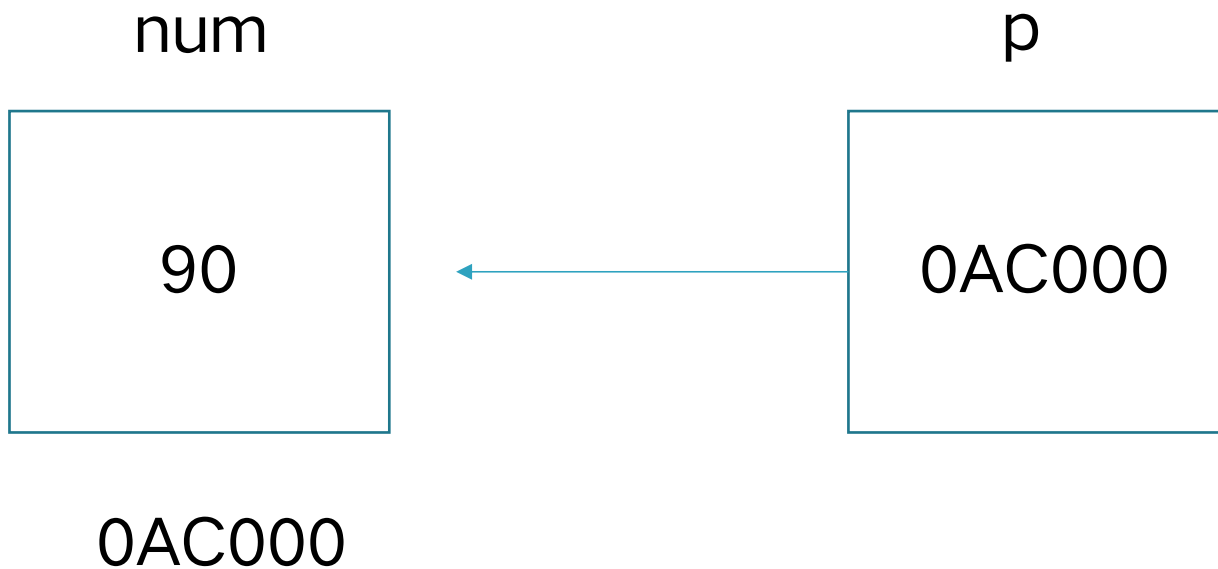
3



## ▶ 5.2 变量的指针和指向变量的指针变量

变量的指针就是变量的地址。

存放变量地址的变量是指针变量。



地址指针的基本概念

指针



变量的指针和指向变量的指针变量

指针真题演练

3

## ▶ 5.2.1 指针变量的定义

指针变量定义格式：

类型说明符 \*变量名;                    /\*表示这是一个指针变量

int \*p;

**注意：指针变量只能指向同类型的变量**



➡ 指针变量的定义

变量的指针和指向变量的指针变量

指针变量的引用

指针变量作为函数参数

## ▶ 5.2.2 指针变量的引用



指针变量的定义

变量的指针和指向变量的指针变量

▶ 指针变量的引用

指针变量作为函数参数

未经赋值的指针变量不可使用，否则造成系统混乱，甚至死机。

关于指针变量赋值与引用相关的运算符：

1) &：取地址运算符

2) \*：指针运算符 //定义变量时表示指针变量；语句中表示取所指向变量的内容

指针变量初始化	指针变量赋值
<code>int a; int *p = &amp;a;</code>	<code>int a, *p; p = &amp;a;</code>

## 课堂案例



```
int x = 3, y, *px = &x;
```

```
y = *px + 5;
```

```
y = ++*px;
```

```
y = *px++;
```

► 5.2.3 指针变量作为函数参数



指针变量的定义

指针变量的引用

→ 指针变量作为函数参数

函数参数不仅可以是整型、实型、字符型等数据，还可以是指针，它的作用是将一个变量的地址传递到另一个函数中，共享主调函数内的变量。

变量的指针和指向变量的指针变量

	值传递	地址传递
实现方式	变量做参数	数组或指针做参数
数据空间	相当于复制粘贴，单独存在	数据空间共享
数据特点	你变我不变	你变我也变

## 课堂案例



- 1、输入两个整数按大小顺序输出。用函数处理，且用指针类型的数据做函数参数。

## ▶ 指针真题演练

广东工业大学 2019年

13. 已有定义：int k=2; int \*ptr1, \*ptr2;，且ptr1和ptr2 均已指向变量k，下面不能正确执行的赋值语句是\_\_\_\_\_.

A k=\*ptr1 + \*ptr2;

B ptr2=k;

C ptr1=ptr2;

D k=\*ptr1 \* (\*ptr2);



地址指针的基本概念

指针

变量的指针和指向变量的指针变量



指针真题演练

3

## ▶ 指针真题演练

广东工业大学 2019年

13.已有定义：int k=2;int\*ptr1, \*ptr2;, 且ptr1和ptr2 均已指向变量k, 下面不能正确执行的赋值语句是\_\_B\_\_.

A k=\*ptr1 + \*ptr2;

B ptr2=k;

C ptr1=ptr2;

D k=\*ptr1\*(\*ptr2);



地址指针的基本概念

指针

变量的指针和指向变量的指针变量



指针真题演练

3



## ▶ 指针真题演练

广东工业大学 2019年

17. 下面程序段执行的结果是\_\_\_\_\_.

```
char *s="hello";
```

```
s++; printf("%d",s);
```

A ello

B e

C 字符 'e' 的地址

D 无确定输出结果



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练

## ▶ 指针真题演练

广东工业大学 2019年

17.下面程序段执行的结果是\_\_C\_\_.

```
char *s="hello";
```

```
s++; printf("%d",s);
```

A ello

B e

C 字符 'e' 的地址

D 无确定输出结果



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练

## ▶ 指针真题演练

广东工业大学 2019年

3.有以下程序：

```
main( ){  
    int *p1,*p2,*p;  
    int a=6, b=12;  
    p1=&a; p2=&b;  
    if (a<b)  
        {p=p1;p1=p2;p2=p;}  
    printf("x=%d,y=%d\n",*p1,*p2);  
}
```

程序运行后的输出结果是什么？（4分）



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练

## ▶ 指针真题演练

广东工业大学 2019年

3.有以下程序：

```
main( ){  
    int *p1,*p2,*p;  
    int a=6, b=12;  
    p1=&a; p2=&b;  
    if (a<b)  
        {p=p1;p1=p2;p2=p;}  
    printf("x=%d,y=%d\n",*p1,*p2);  
}
```

程序运行后的输出结果是什么？（4分）

答案：12 ， 6



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3

指针真题演练

## ▶ 指针真题演练

广东工业大学 2019年

4.有以下程序：程序运行后的输出结果是什么？（4分）

```
#include "stdio.h"
```

```
#define FUN(x,y) (x+2)*y
```

```
sub(int *a, int *b)
```

```
{
```

```
    int t;
```

```
    t=*a+*b;
```

```
    *a=t-*a;
```

```
    *b=t-*b;
```

```
}
```

```
int main()
```

```
{
```

```
    int x=3, y=4, s;
```

```
    sub(&x,&y);
```

```
    s=36/FUN(x,y);
```

```
    printf("s=%d"s);
```

```
}
```

指针

地址指针的基本概念

变量的指针和指向变量的指针变量

指针真题演练



3

## ▶ 指针真题演练

广东工业大学 2019年

4.有以下程序：程序运行后的输出结果是什么？（4分）

```
#include "stdio.h"
```

```
#define FUN(x,y) (x+2)*y
```

```
sub(int *a, int *b)
```

```
{
```

```
    int t;
```

```
    t=*a+*b;
```

```
    *a=t-*a;
```

```
    *b=t-*b;
```

```
}
```

```
int main()
```

```
{
```

```
    int x=3, y=4, s;
```

```
    sub(&x,&y);
```

```
    s=36/FUN(x,y);
```

```
    printf("s=%d"s);
```

```
}
```

答案： 18



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练

## ▶ 指针真题演练

广东工业大学 2019年

5.有以下程序：

```
#include <stdio.h>
char *p = "abcdefghijklmnopq";
main(){
    int i=0;
    while(*p++!='e');
    printf("%c\n", *p);
}
```

程序运行后的输出结果是什么？（4分）



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练

## ▶ 指针真题演练

广东工业大学 2019年

5.有以下程序：

```
#include <stdio.h>
char *p = "abcdefghijklmnopq";
main(){
    int i=0;
    while(*p++!='e');
    printf("%c\n", *p);
}
```

程序运行后的输出结果是什么？（4分）

答案：f



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练



## ▶ 指针真题演练

北京航空航天大学 2018年

5.若有以下变量的声明语句:

```
int i = 1,a[]={0,2,4},*b;
```

```
b=&i;
```

则下列选项中, 其结果与表达式 “\* (a + 1)”相等的是( ).

A.a[0];

B.\*a+i;

C.\*(a + b);

D.\*(a+\*b)。



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练

## ▶ 指针真题演练

北京航空航天大学 2018年

5.若有以下变量的声明语句:

```
int i = 1,a[]={0,2,4},*b;
```

```
b=&i;
```

则下列选项中, 其结果与表达式 “ $* (a + 1)$ ”相等的是( **D** ).

A.a[0];

B.\*a+i;

C.\*(a + b);

D.\*(a+\*b)。



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

3



指针真题演练

## ► 指针真题演练

北京航空航天大学 2018年

2.若分别采用下列两种不同的方式声明字符串变量msg1和msg2,

`char*msg1; char msg2[20];`

则请问：这两种声明方式有什么不同?使用它们时分别需要注意什么?



地址指针的基本概念

指针

变量的指针和指向变量的指针变量



指针真题演练

3

## ▶ 指针真题演练

北京航空航天大学 2018年

2.若分别采用下列两种不同的方式声明字符串变量msg1和msg2,

`char*msg1; char msg2[20];`

则请问：这两种声明方式有什么不同?使用它们时分别需要注意什么？

(1) \*msg1表示定义一个变量，占一个字节，可指向一个字符串的收地址；

msg2定义一个数组，占20个字节，数据可变。

(2) \*msg1定义后只能初始化或直接赋值，不可随机输入，否则可能会警告；

msg2只可存储19个字符的字符串。



地址指针的基本概念

指针

变量的指针和指向变量的指针变量



指针真题演练

③

## ► 指针真题演练



北京航空航天大学 2021年

7. 下列程序的功能是判断输入的字符串是否是“回文”。(注：按顺序读与按逆序读都一样的字符串被称为“回文”，例如：abcdcba)。请在程序的空白处(符号-----处)填入合适内容，使程序完整。

```
int main( ){
    char ch[81], *p =ch, *q;
    gets(p);
    q =p+-----;
    while(-----){
        if(*p ==*q) {p++; q--;}
        else break;
    }
    if(p<q)
        printf( “该字符串不是回文! \n” );
    else
        printf( “该字符串是回文! \n” );
}
```

指针

地址指针的基本概念

变量的指针和指向变量的指针变量



指针真题演练

③

## ▶ 指针真题演练



地址指针的基本概念

指针

变量的指针和指向变量的指针变量

指针真题演练

③

北京航空航天大学 2021年

7. 下列程序的功能是判断输入的字符串是否是“回文”。(注：按顺序读与按逆序读都一样的字符串被称为“回文”，例如：abcdcba)。请在程序的空白处(符号-----处)填入合适内容，使程序完整。

```
int main( ){
    char ch[81], *p =ch, *q;
    gets(p);
    q =p+ strlen(p)-1;
    while( p < q ){
        if(*p ==*q) {p++; q--;}
        else break;
    }
    if(p<q)
        printf( “该字符串不是回文! \n” );
    else
        printf( “该字符串是回文! \n” );
}
```

## ► 指针真题演练

北京航空航天大学 2021年

### 七、 程序设计题 （此题 15 分）

请编写一程序， 该程序的功能是找出而且删除一维整型数组  $a[100]$  中的最小值元素。 要求：

1. 数组各元素通过键盘输入获得初值；
2. 所有对数组元素的引用必须通过指针完成。



地址指针的基本概念

指针

变量的指针和指向变量的指针变量



指针真题演练

3

## ▶ 指针真题演练

北京师范大学 2014年

3.写出下列程序的输出结果(5分)

```
func(int *s,int *y){
    static int t=3;
    *y=s[t];
    t--;
}
main(){
    int a[]={10,20,30,40},i,x=0;
    for(i=0;i<4;i++){
        func(a,&x);
        printf( "%d" ,x);
    }
    printf( "\n" );
}
```



地址指针的基本概念

指针

变量的指针和指向变量的指针变量



指针真题演练

3



## ▶ 指针真题演练

北京师范大学 2014年

3.写出下列程序的输出结果(5分)

```
func(int *s,int *y){
    static int t=3;
    *y=s[t];
    t--;
}
main(){
    int a[]={10,20,30,40},i,x=0;
    for(i=0;i<4;i++){
        func(a,&x);
        printf( "%d" ,x);
    }
    printf( "\n" );
}
```

答案: 40 30 20 10



地址指针的基本概念

指针

变量的指针和指向变量的指针变量



指针真题演练

3

## ▶ 课堂总结



### 1、指针运算符

	取地址运算符&	取内容运算符*
操作数个数	单目运算符	单目运算符
结合性	自右至左	自右至左
举例	++*p	

### 2、指针变量的运算

#### 1) 赋值运算

例如: `int a, *pa; pa = &a;` 或 `int a[5], *pa; pa = a;`

#### 2) 加减算术运算

针对于指向数组的指针变量, 加或减整数 $n$ , 表示指针从当前指向位置向前或向后移动 $n$ 个位置。

#### 3) 两个指针变量之间的运算

只针对指向同一数组的两个指针变量之间做运算, 否则运算无意义。指针变量相减获得两个指针所指数组元素之间相差的元素个数。

## 4.1 ASCII码值



常用字符与 ASCII 代码对照表

为了便于查询，以下列出 **ASCII 码表**：第 128~255 号为扩展字符（不常用）

ASCII 码	键盘	ASCII 码	键盘	ASCII 码	键盘	ASCII 码	键盘
27	ESC	32	SPACE	33	!	34	"
35	#	36	\$	37	%	38	&
39	'	40	(	41	)	42	*
43	+	44	,	45	-	46	.
47	/	48	0	49	1	50	2
51	3	52	4	53	5	54	6
55	7	56	8	57	9	58	:
59	;	60	<	61	=	62	>
63	?	64	@	65	A	66	B
67	C	68	D	69	E	70	F

## 4.1 ASCII码值



71↵	G↵	72↵	H↵	73↵	I↵	74↵	J↵
75↵	K↵	76↵	L↵	77↵	M↵	78↵	N↵
79↵	O↵	80↵	P↵	81↵	Q↵	82↵	R↵
83↵	S↵	84↵	T↵	85↵	U↵	86↵	V↵
87↵	W↵	88↵	X↵	89↵	Y↵	90↵	Z↵
91↵	[↵	92↵	\↵	93↵	]↵	94↵	^↵
95↵	_↵	96↵	`↵	97↵	a↵	98↵	b↵
99↵	c↵	100↵	d↵	101↵	e↵	102↵	f↵
103↵	g↵	104↵	h↵	105↵	i↵	106↵	j↵
107↵	k↵	108↵	l↵	109↵	m↵	110↵	n↵
111↵	o↵	112↵	p↵	113↵	q↵	114↵	r↵
115↵	s↵	116↵	t↵	117↵	u↵	118↵	v↵
119↵	w↵	120↵	x↵	121↵	y↵	122↵	z↵
123↵	{↵	124↵	↵	125↵	}↵	126↵	~↵

## ► 字符ASCII码问题

广东工业大学 2019年

9.若有定义：char c='\010'，则变量c中包含的字符个数为\_\_\_\_\_。



2.1 开关控制 ⊕

真题讲解 ⊕

3.1 ASCII码值

真题讲解 ⊖

字符ASCII码问题

4.1 数学运算 ⊕

真题讲解 ⊕

真题演练-2 ⊖

## ► 字符ASCII码问题

广东工业大学 2019年

9.若有定义：char c='\010'，则变量c中包含的字符个数为\_\_\_1\_\_\_。

'\010'：代表字符个数是1个。

因为\010是转义字符，即'\010'只代表一个ASCII值为8（十进制）的字符。'\010'中第一个0表示是以八进制计算的。



2.1 开关控制 ⊕

真题讲解 ⊕

3.1 ASCII码值

真题讲解 ⊖

字符ASCII码问题

4.1 数学运算 ⊕

真题讲解 ⊕

真题演练-2 ⊖

## ► 字符ASCII码问题

3、在执行以下程序时，如果从键盘上输入：ABCdef，则输出为（ ）。

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

```
void main( ){
```

```
    char ch;
```

```
    while ((ch=getchar( ))!='\n'){
```

```
        if (ch>='A' && ch<='Z')
```

```
            ch=ch+32;
```

```
        else if (ch>='a' && ch<='z')
```

```
            ch=ch-32;
```

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

```
    }
```

```
    printf("\n");
```

```
}
```

(A) ABCdef

(C) abc

(B) abcDEF

(D) DEF



2.1 开关控制 ⊕

真题讲解 ⊕

3.1 ASCII码值

真题讲解 ⊖

字符ASCII码问题

4.1 数学运算 ⊕

真题讲解 ⊕

真题演练-2 ⊖



## ► 字符ASCII码问题

3、在执行以下程序时，如果从键盘上输入：ABCdef，则输出为（ B ）

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

```
void main( ){
```

```
    char ch;
```

```
    while ((ch=getchar( ))!='\n'){
```

```
        if (ch>='A' && ch<='Z')
```

```
            ch=ch+32;
```

```
        else if (ch>='a' && ch<='z')
```

```
            ch=ch-32;
```

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

```
    }
```

```
    printf("\n");
```

```
}
```

(A) ABCdef

(C) abc

(B) abcDEF

(D) DEF

真题演练-2

2.1 开关控制

真题讲解

3.1 ASCII码值

真题讲解

字符ASCII码问题

4.1 数学运算

真题讲解



## ► 字符ASCII码问题

5. 下列程序的功能是将小写字母转换成对应的大写字母后的第 2 个字母，例如，将 a 转换成 C，将 b 转换成 D，其中，y 转换成 A，z 转换成 B。请在程序的空白处(符号-----处)填入合适内容，使程序完整。

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

```
main( ){
```

```
    char ch;
```

```
    while((ch=getchar( ))!= '\n' )
```

```
        if(ch>= 'a' && ch<= 'z' ){
```

```
            _____;
```

```
            if(ch> 'Z' && ch<= 'Z' +2)
```

```
                _____;
```

```
        }
```

```
    }
```



2.1 开关控制 ⊕

真题讲解 ⊕

3.1 ASCII码值

真题讲解 ⊖

字符ASCII码问题

4.1 数学运算 ⊕

真题讲解 ⊕

真题演练-2 ⊖

## ► 字符ASCII码问题

5. 下列程序的功能是将小写字母转换成对应的大写字母后的第 2 个字母，例如，将 a 转换成 C，将 b 转换成 D，其中，y 转换成 A，z 转换成 B。请在程序的空白处(符号-----处)填入合适内容，使程序完整。

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

```
main( ){
```

```
    char ch;
```

```
    while((ch=getchar( ))!= '\n' )
```

```
        if(ch>= 'a' && ch<= 'z' ){
```

```
            ch = ch - 32 + 2 ;
```

```
            if(ch> 'Z' && ch<= 'Z' +2)
```

```
                ch = ch -26 ;
```

```
        }
```

```
    }
```



2.1 开关控制 ⊕

真题讲解 ⊕

3.1 ASCII码值

真题讲解 ⊖

字符ASCII码问题

4.1 数学运算 ⊕

真题讲解 ⊕

真题演练-2 ⊖

## ► 字符ASCII码问题

4.已知数字0的ASCII码为48，请写出下列程序的正确输出。

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

```
int main ( ) {
```

```
    char c=48;
```

```
    int i,mask=01;
```

```
    for ( i = 1;i<=5; i++) {
```

```
        printf ( "%c",c | mask) ;
```

```
        mask=mask<<1;
```

```
    }
```

```
    return 0;
```

```
}
```



2.1 开关控制 ⊕

真题讲解 ⊕

3.1 ASCII码值

真题讲解 ⊖

字符ASCII码问题

4.1 数学运算 ⊕

真题讲解 ⊕

真题演练-2 ⊖

## ► 字符ASCII码问题

4.已知数字0的ASCII码为48，请写出下列程序的正确输出。

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

```
int main ( ) {
```

```
    char c=48;
```

```
    int i,mask=01;
```

```
    for ( i = 1;i<=5; i++) {
```

```
        printf ( "%c",c | mask) ;
```

```
        mask=mask<<1;
```

```
    }
```

```
    return 0;
```

```
}
```

*i=1 : '1'; i=2 : '2'; i=3 : '4'; i=4 : '8'; i=5 : '0';*



2.1 开关控制 ⊕

真题讲解 ⊕

3.1 ASCII码值

真题讲解 ⊖

字符ASCII码问题

4.1 数学运算 ⊕

真题讲解 ⊕

真题演练-2 ⊖

# 预祝金榜题名，成功上岸！



北京开课吧科技有限公司  
[www.kaikeba.com](http://www.kaikeba.com)  
400-996-0826

## Thanks