设a=5,b=6,c=7,d=8,m=2,n=2，执行(m=a > b）&&(n=c>d)后n的值为2。对

a>b为假,得0,再赋值给m=0

又因为&&前面为0就不执行后面

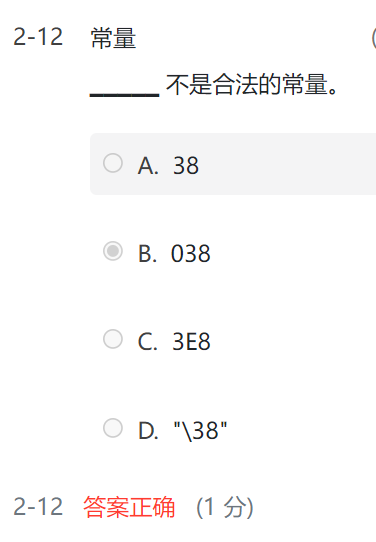
执行以下程序段，输入10，输出10.00。错

double x;

scanf("%d", &x); 使其变为整数型

printf("%.2f", x);

结果是出错，因为scanf中前后类型不对应

设变量已正确定义，执行以下程序段，顺序输入三个字符'k'，则输出k。 对

scanf("%c%c%c", &ch1, &ch2, &ch3);

printf("%c", ch2);

038以0开头应当为八进制但是8超出了八进制

表达式 !!6的值是6。 错

！是逻辑运算符，结果要么是0，要么是1。！6=0，！0=1

双引号是字符串

单引号的是字符

**下面程序段的输出是（ ）。** 18

**int x=023; -----这是8进制**

**printf("%d\n",--x);**

执行以下程序段，sum的值是55 错

int i, sum;

for (i = 1; i <= 10; i++){

sum = sum + i;

}

Sum初始值未定义

While内语句也要执行

while为真才执行，不要看反

**关于字符范围判断的写法：**

**if (a[i]>='0'&&a[i]<='9')**

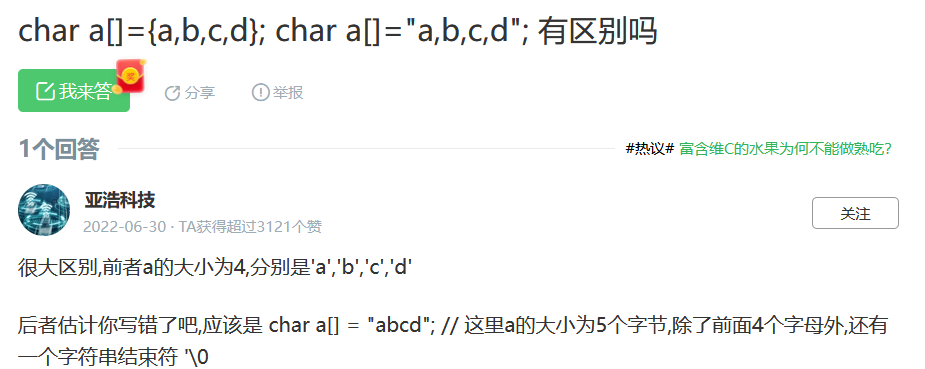
1. 数组长度的计算：整形数组大小：sizeof(数组名)/sizeof(数组类型)。

字符数组大小：strlen(字符串名)+1,1因为要加上'/0'（结束符），所以再加1

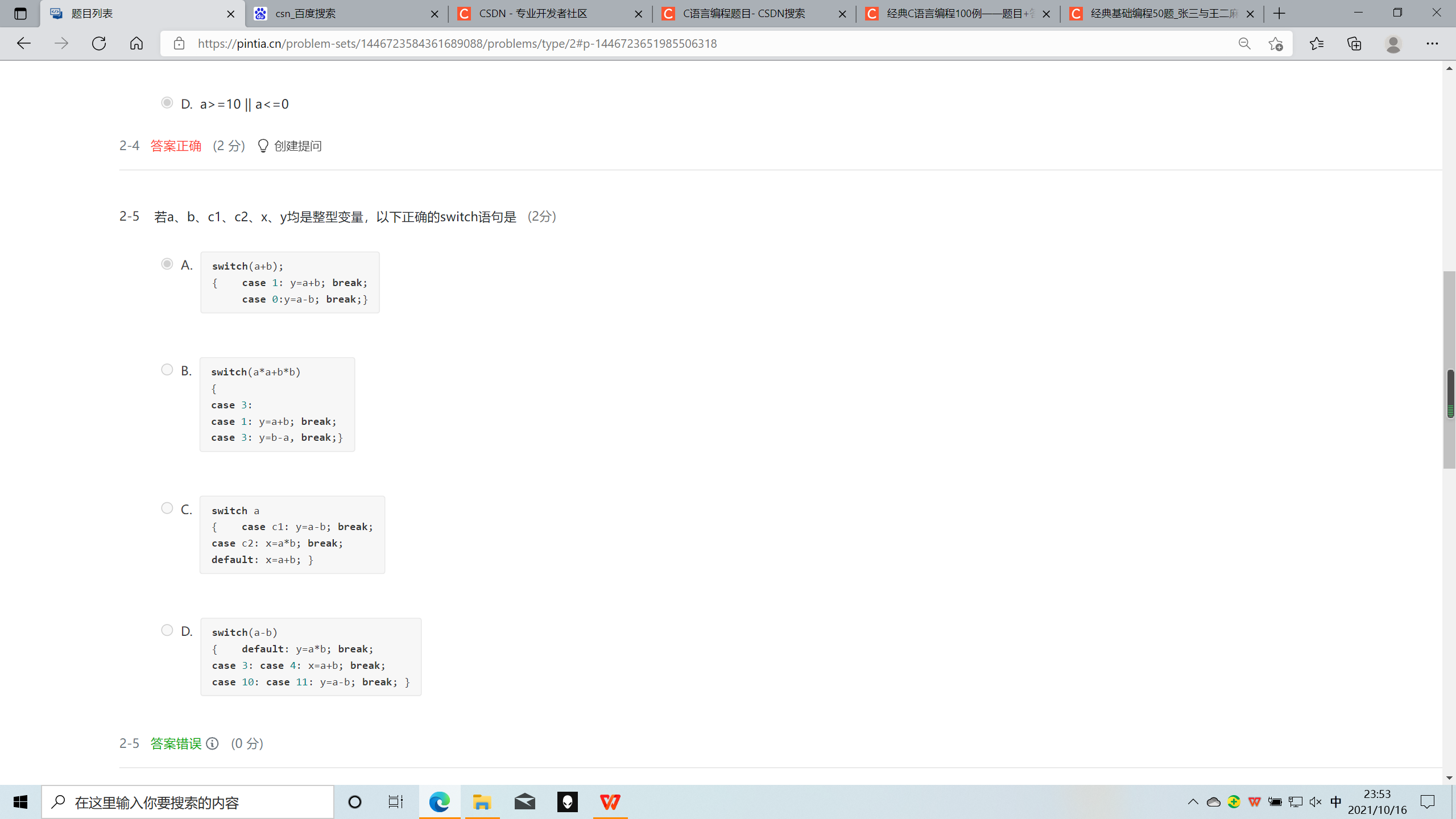
1. 字符串长度的计算：

字符串长度是指字符串所含的字符个数，但不包括最后的'\0'。sizeof(str)

**？“a,b,c,d”与{“a”,”b”,”c”,”d”}区别**

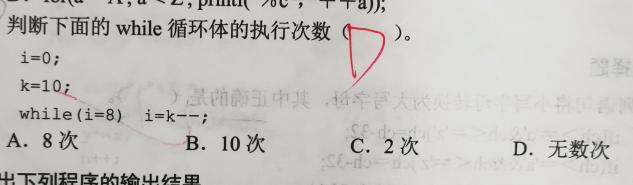
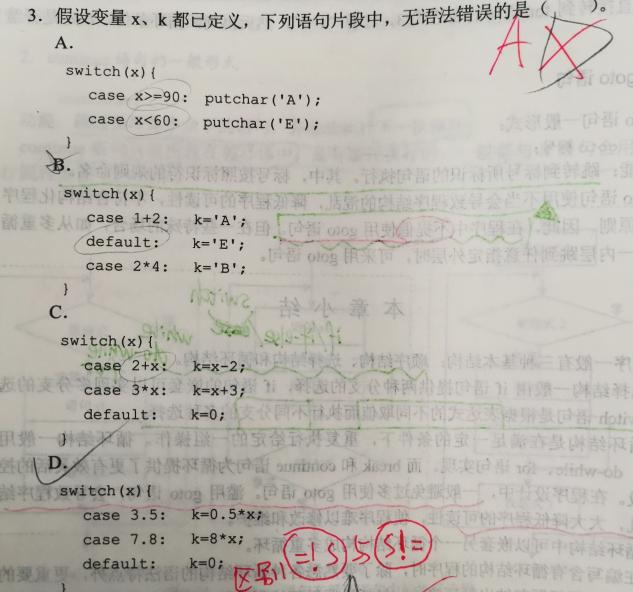
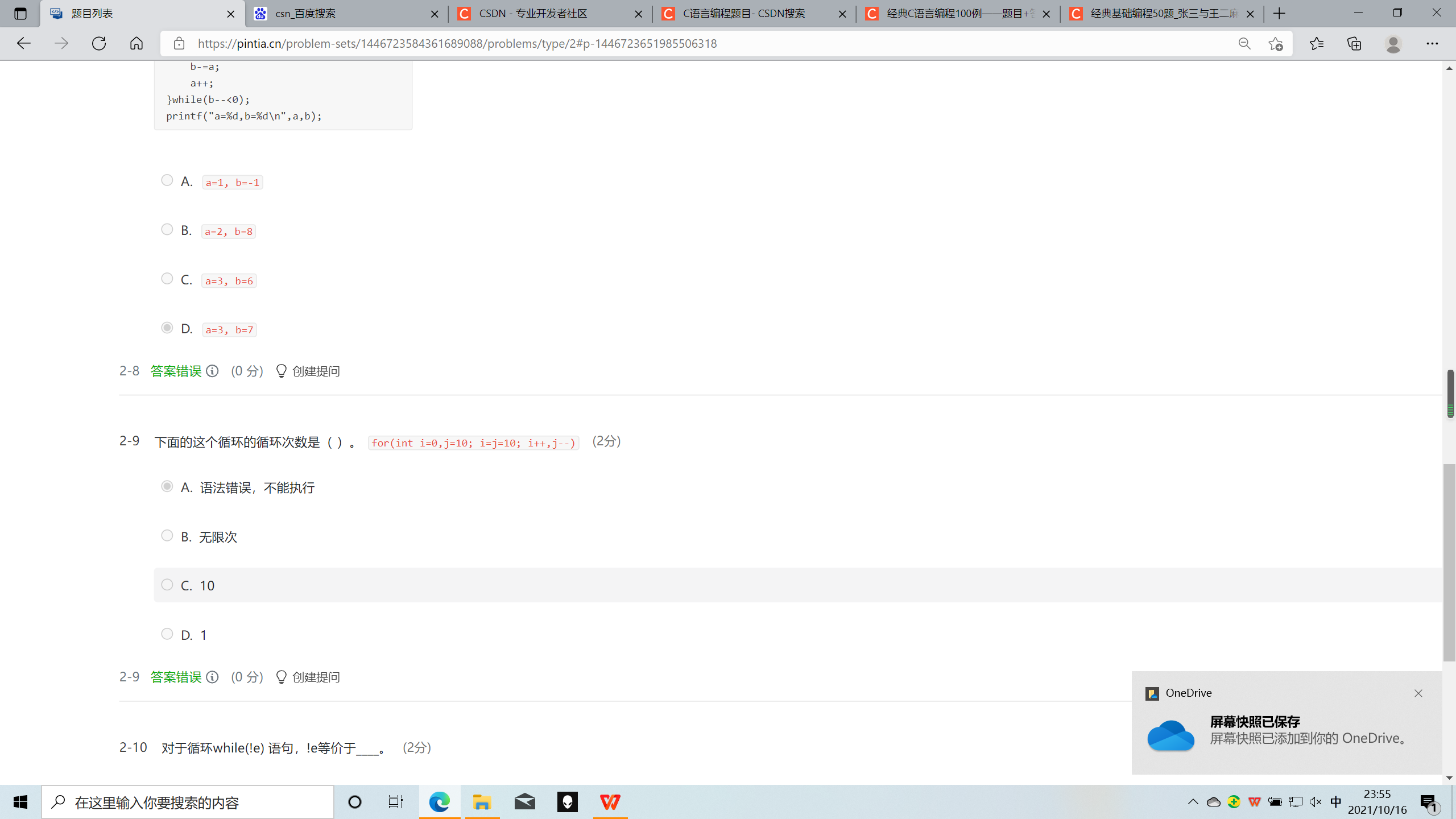


C语言的字符串是由字符数组形式保存的，并约定'\0'(ASCII码值为0)作为字符串结束符。其长度为从字符串开始，到'\0'结束，所有字符的个数，**不包括'\0'本身**。



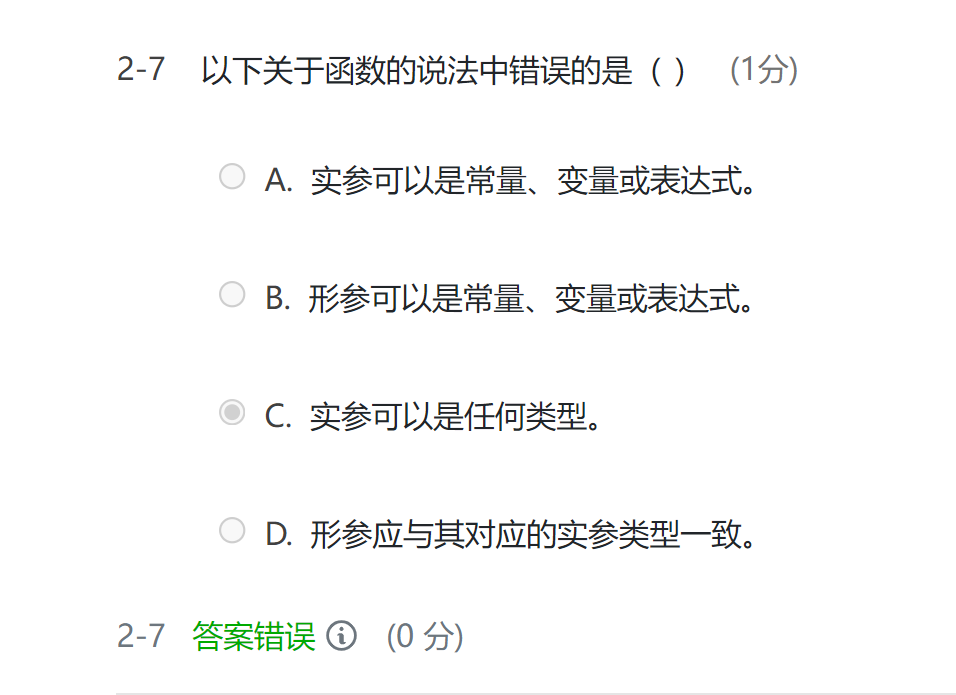
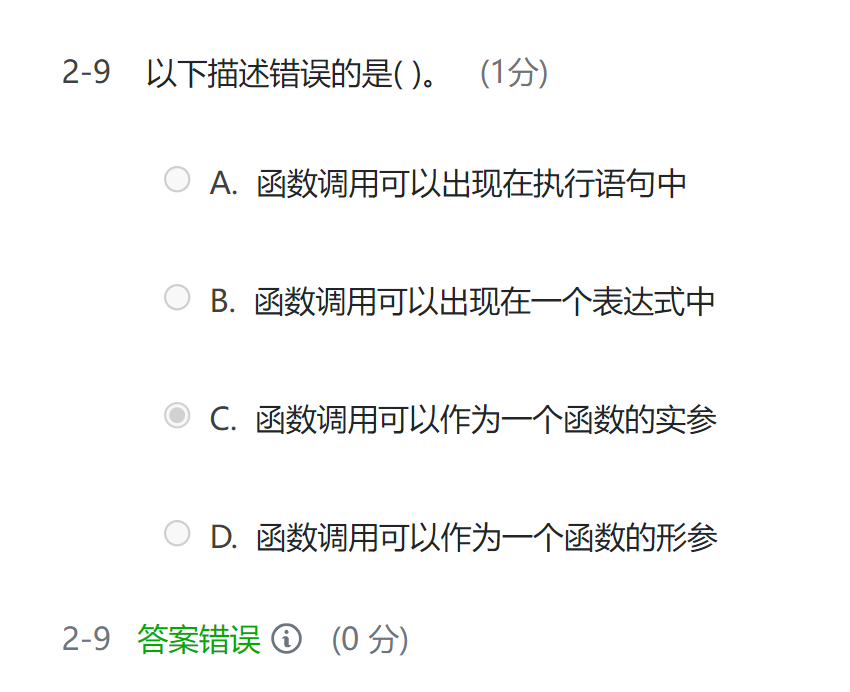
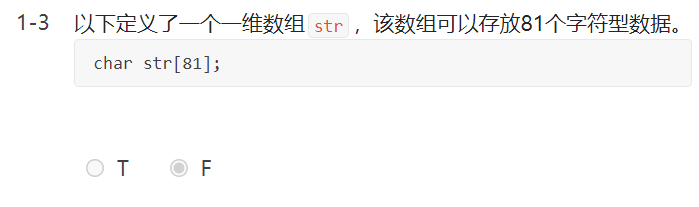
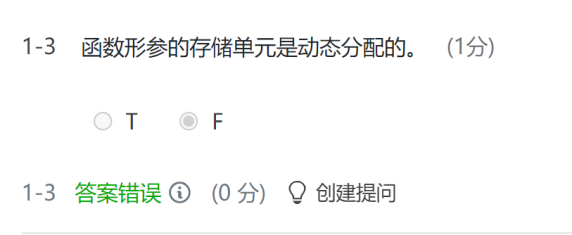
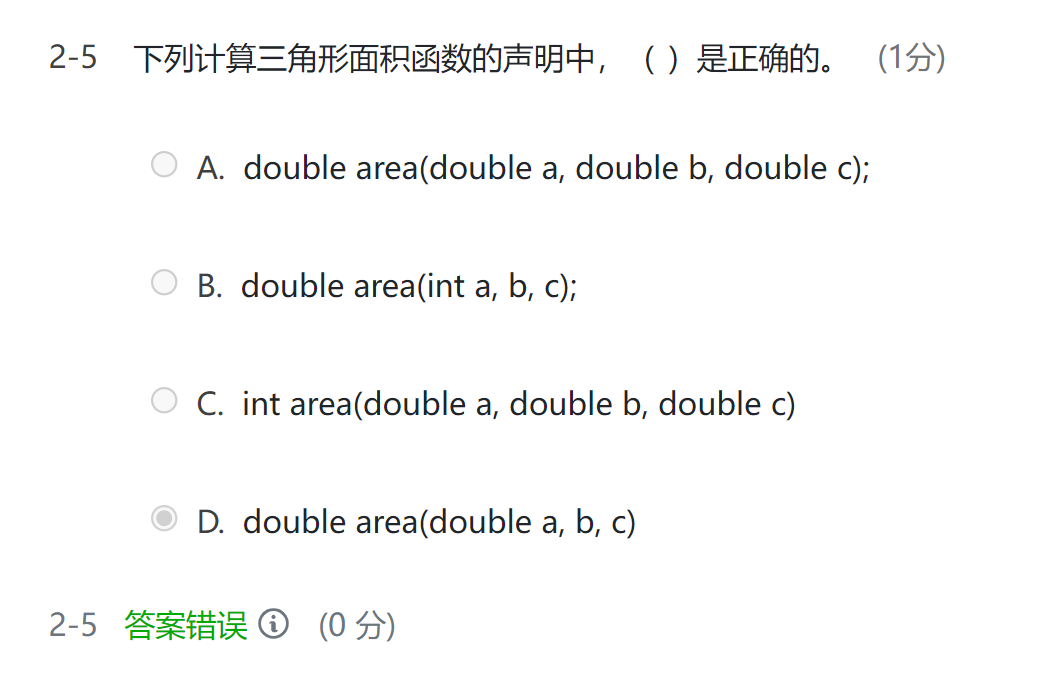
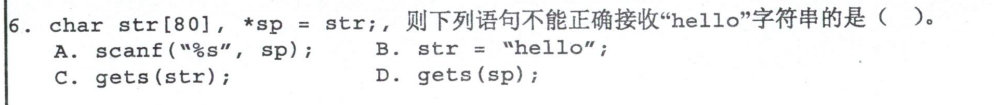
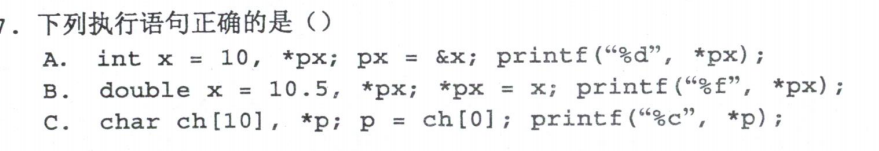
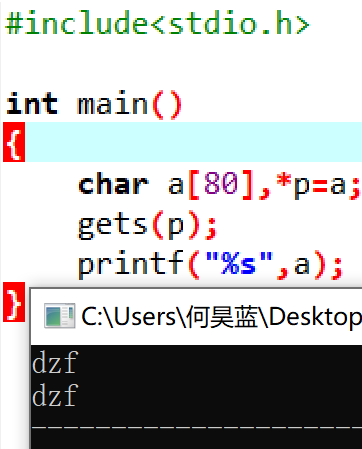
不可在switch（）后加“；”

Case后常量不可重复出现。



注意是 = 还是 ==

这里中间条件语句结果看最左边变量的值，值非零为真，零为假。



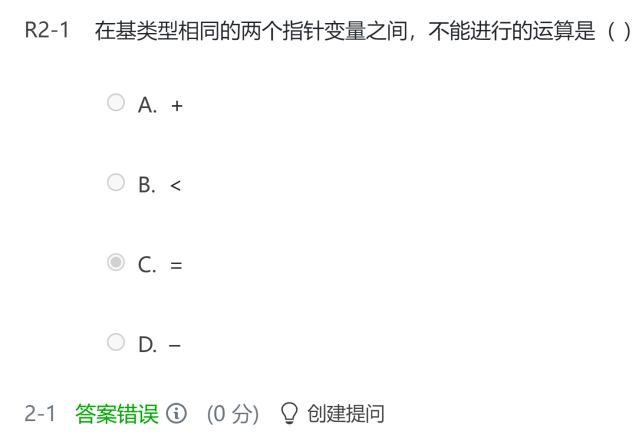
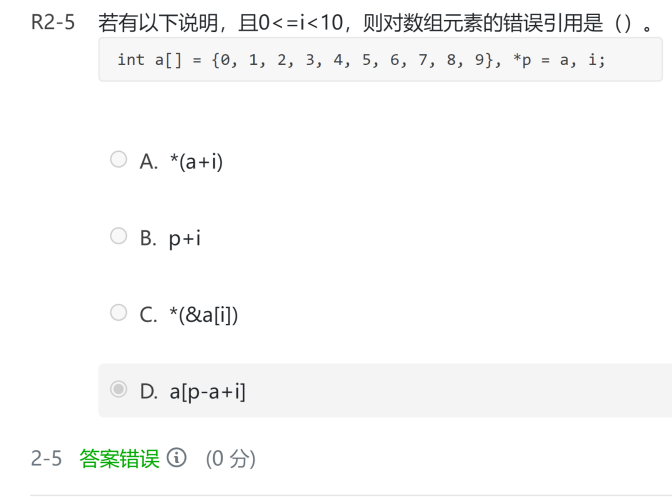
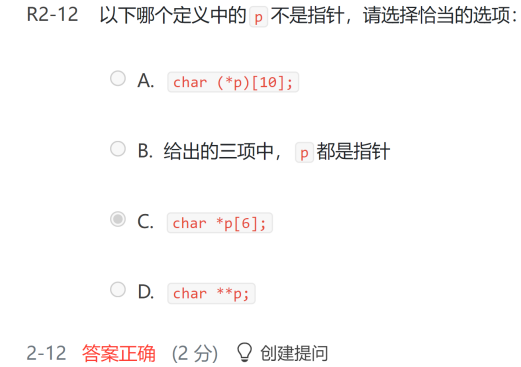
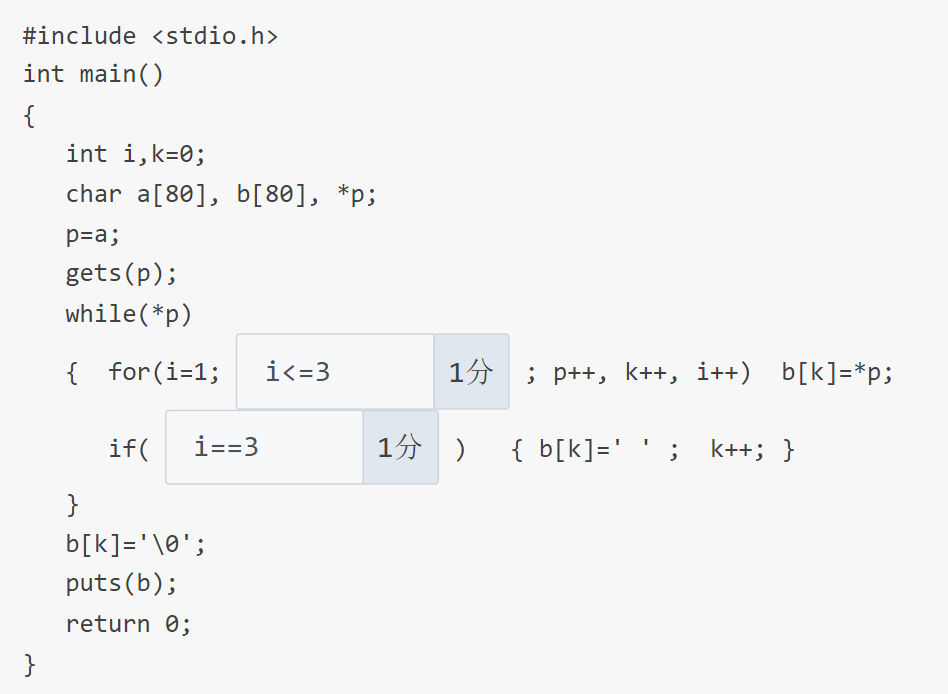
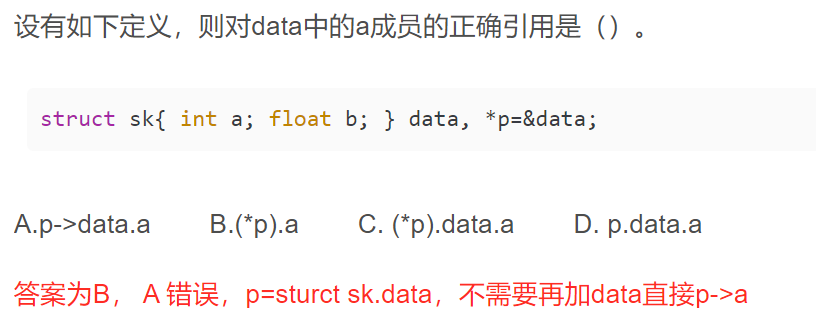
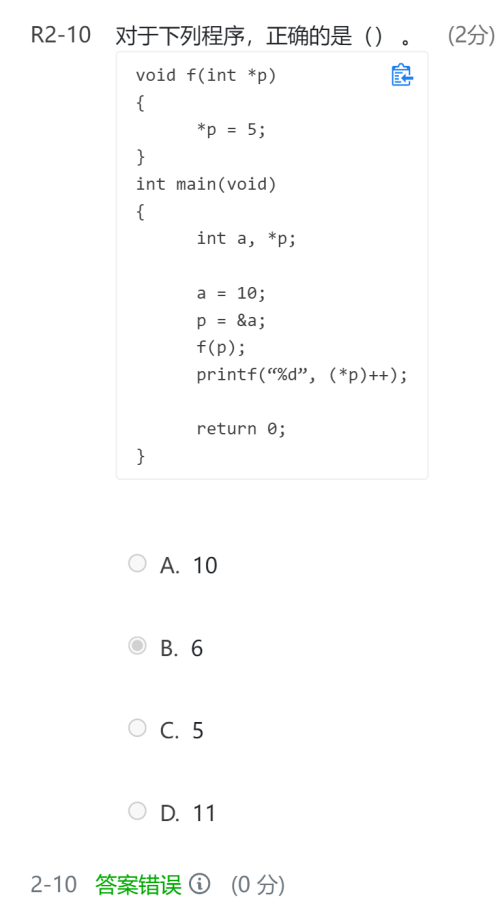
“\0”算什么

NB：A对B错

实参可以是常量、变量、表达式、函数等任意类型，形参只能是变量

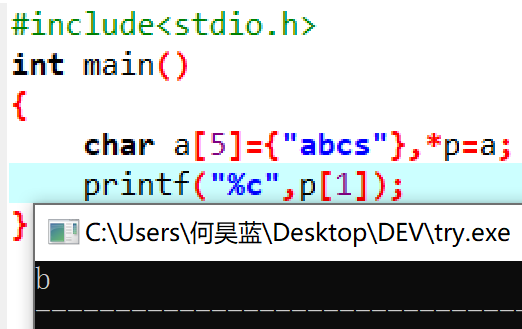
形参在运算过程中会发生改变，但不一定会影响到实参值，所以可知两者不共用内存单元。

“\0”也是字符型

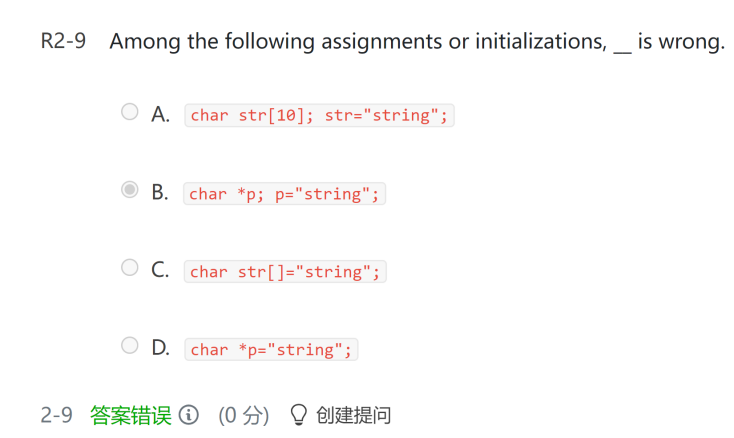
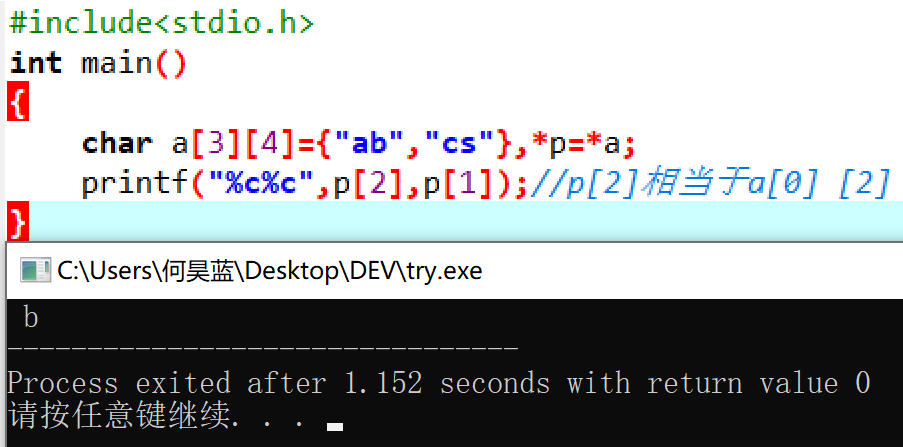


ST data，\*p=&data

则p->a正确



但有



char ch[2][3]={"ab","cd"},\*p=(char \*)ch;（或\*p=\*ch）**正确**

但printf("%s",p+1);输出：b不像printf("%s",ch+1);输出：cd

p是指针p+1即指向下一地址|ch是行指针ch+1代表跳一列

char ch[2][3]={"ab","cd"},\*p;

\*p=(char \*)ch; p=&ch[0]; p=&ch; p=ch; \*p=ch[0][0];错

p=&ch[0][0]; p=ch[0]; \*p=\*ch;**正确**（均输出：ab）

\*(p+2)、\*(ch+3)是别的不知道什么地方的地址了

printf("%s",ch);输出：ab

printf("%s",ch+1);输出：cd ch是行指针

printf("%c",ch);输出出错

A项是取\*++p后值加2在输出ascii对应字符，C同理

B完全没有这种操作

A（数组）分开写错

B（指针）正确

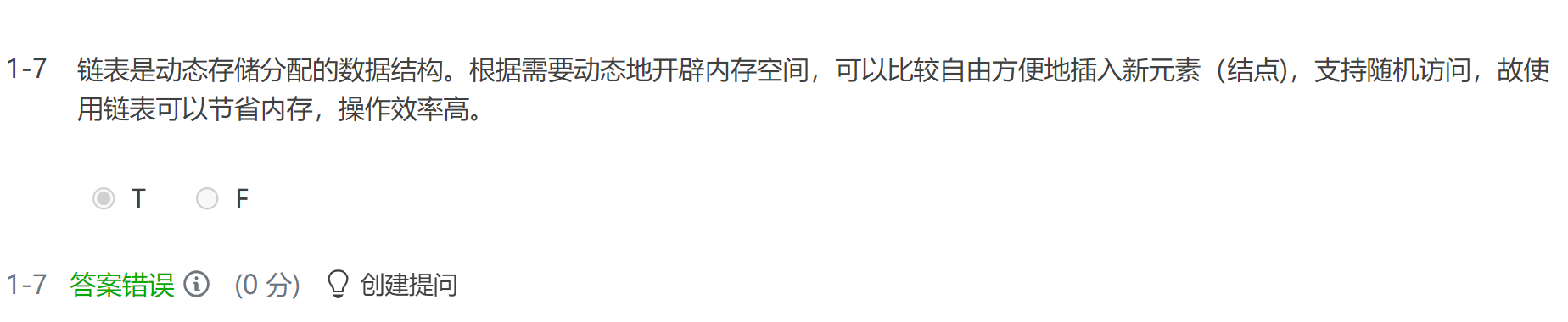
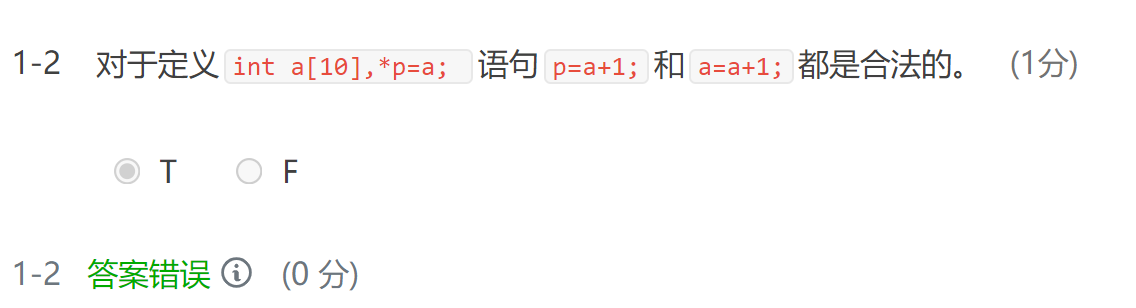
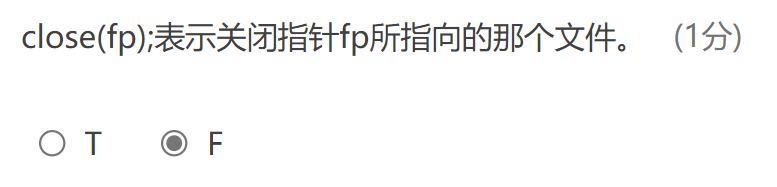
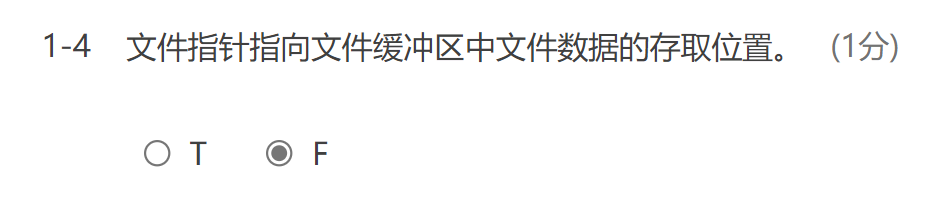
\*p

\*p&&i<=3

B项应该加个“\*”

\*p[]中p是数组

A) + 因为两个指针变量之间可以比较大小 也可以赋值 相减以后得到的使两指针的位移偏量相加,我还真不知道得出来得是什么



P271

P273

**？**当fclose被调用时，一个打开的文件(以文件指针fp标识)会被关闭，关闭后，无法再对fp执行读写操作。



句2不合法，数组名（是地址常量）不可赋值

**？**

**链表不支持随机访问**

不能赋值，但这样可以

++p->str

指针str的值加1

\*p->str

读取指针str所指向的对象的值

\*p->str++

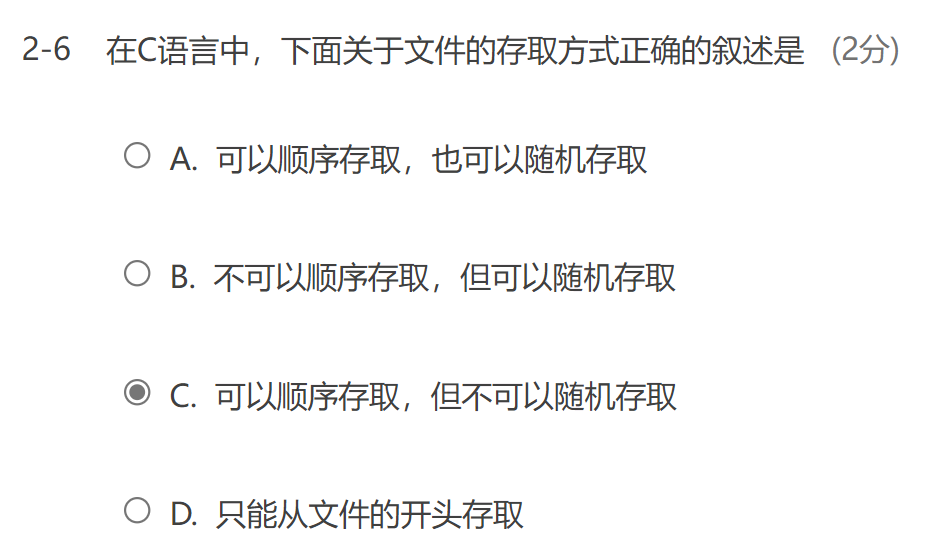
先读取指针str指向的对象的值，然后再将str加1

(\*p->str)++

将指针str指向的对象的值加1

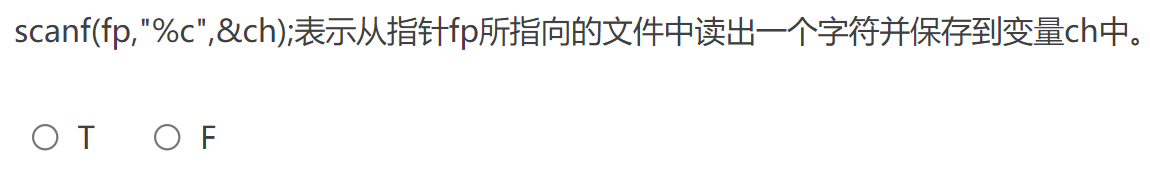
\*p++->str

先读取指针str指向的对象的值，然后p加1

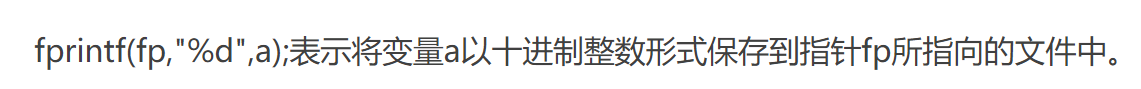


P275

feof()是先读文件在判断是不是到达文件结束，所以当使用feof()作为循环判断条件的话，**就需要注意一种情况，就是当文件为空的时候，它也会读取一次文件内容然后再跳出循环**，所以有可能会读取到一些不可知的内容出来。

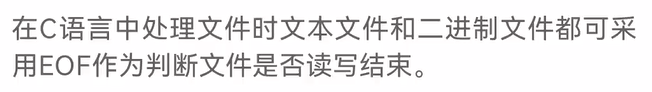
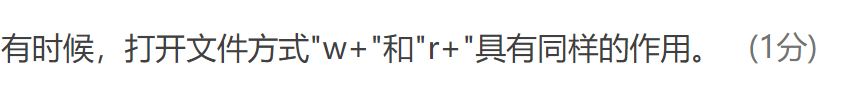


应该是fscanf

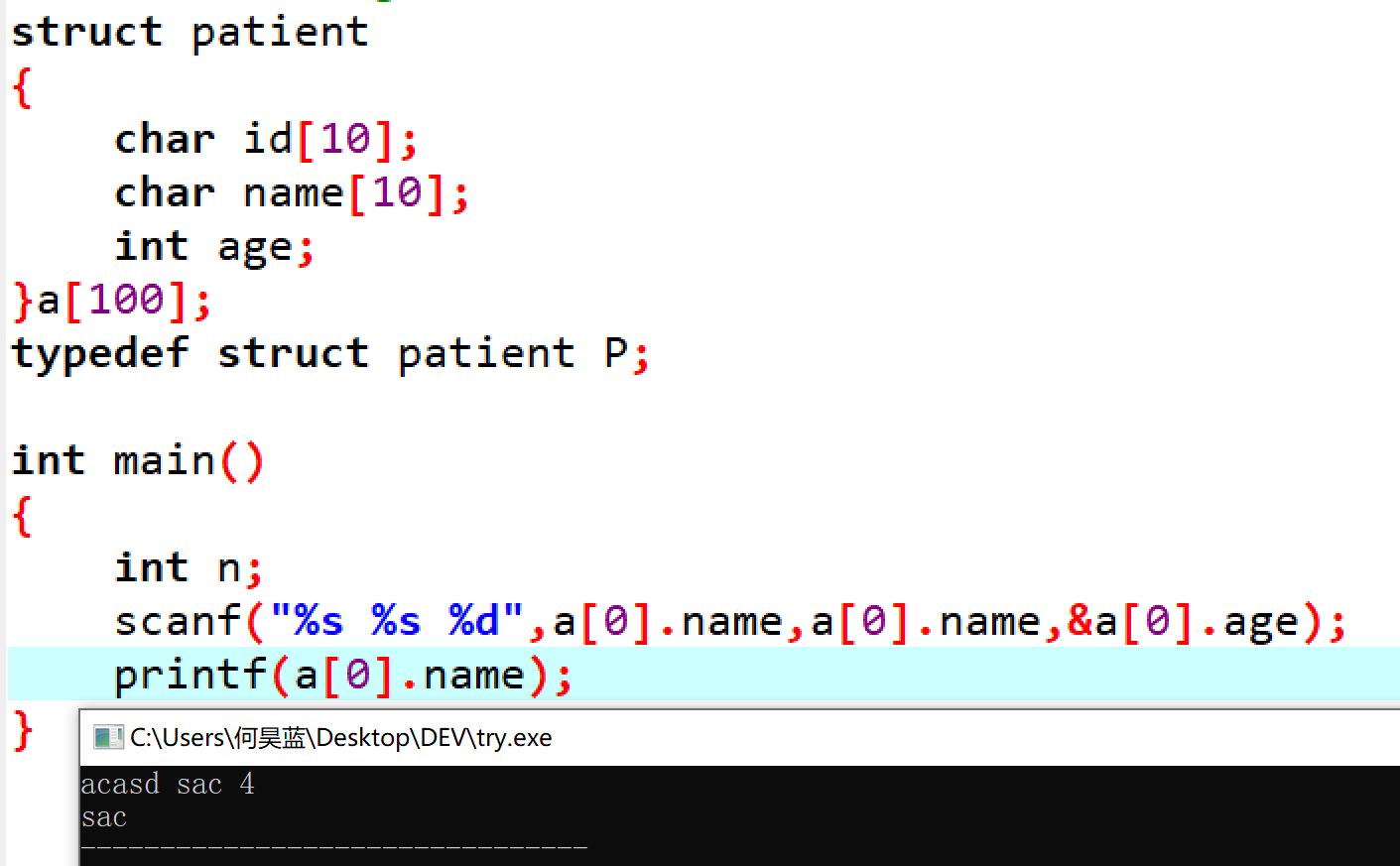


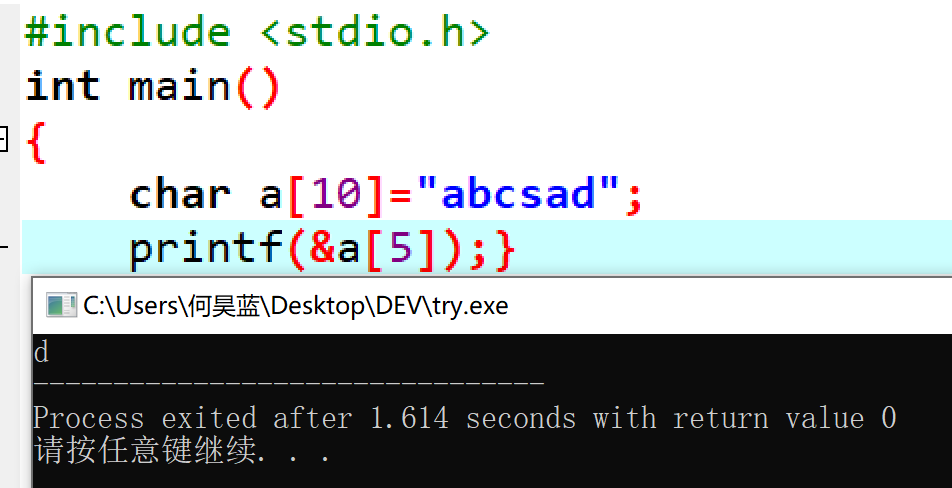
二进制文件不可以用txt作为扩展名，否则会乱码。

但是如果强制把文件后缀改掉，那是可以的

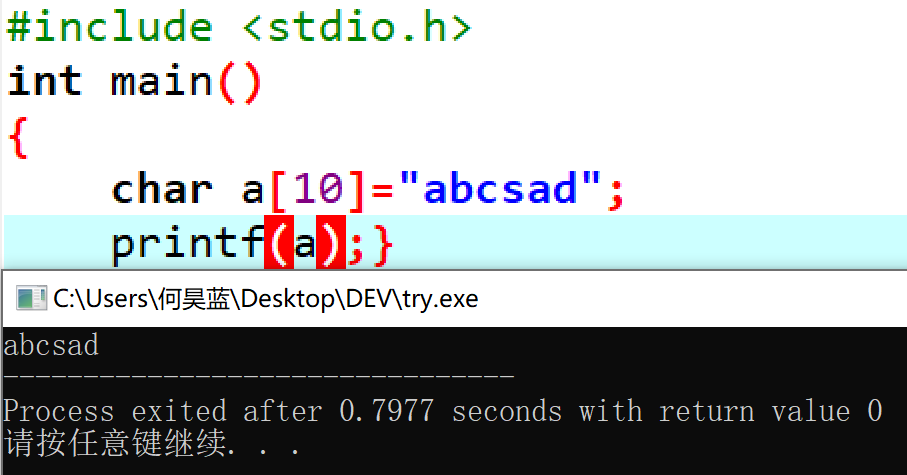


C语言中的文件是流式文件，因此只能顺序存取数据。但是通过fseek函数，可运动文件指针。当对文件的读写操作完成之后，必须将它关闭？，否则可能导致数据丢失。

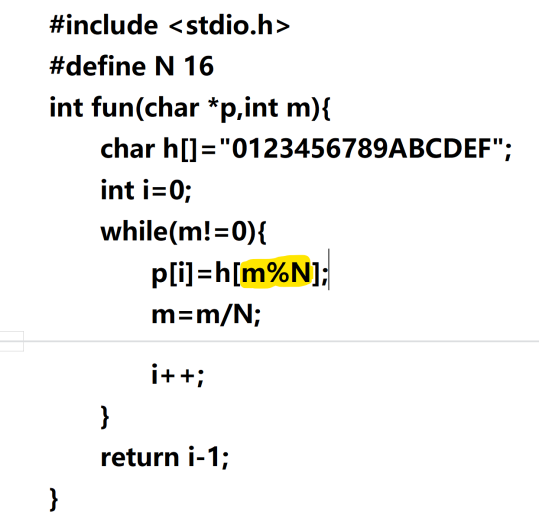


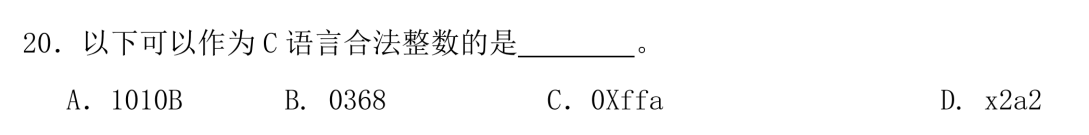


要加&



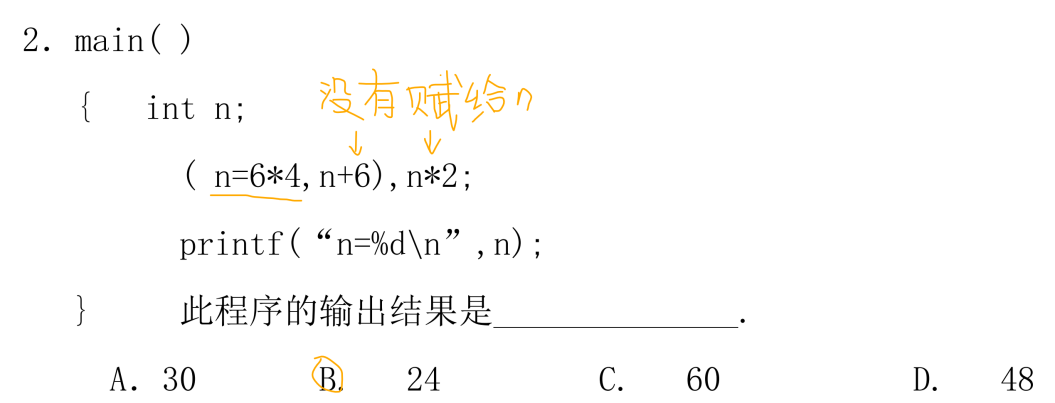
不加&

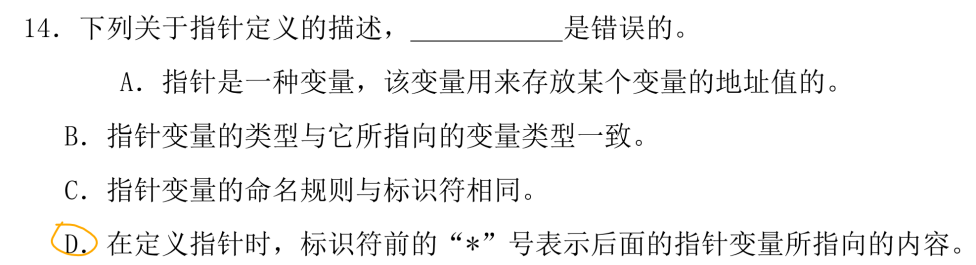




整型常量不包含二进制

只有十（开头不为0）、八（开头必为0）、十六（0x）进制三种

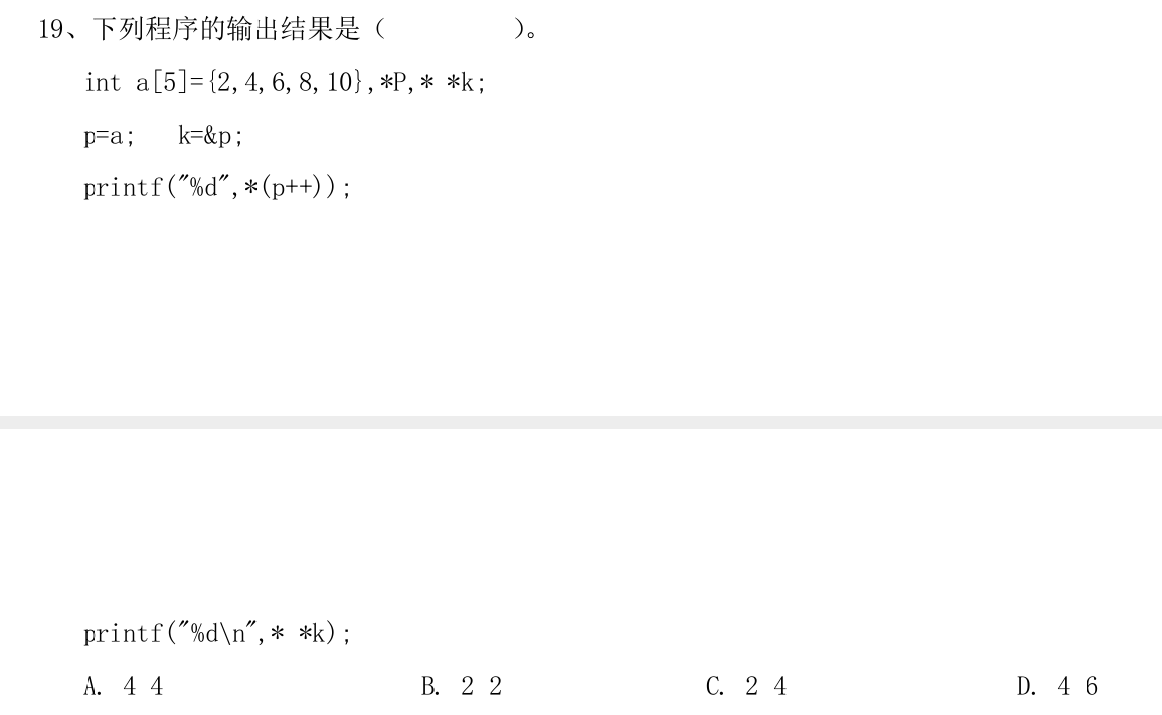
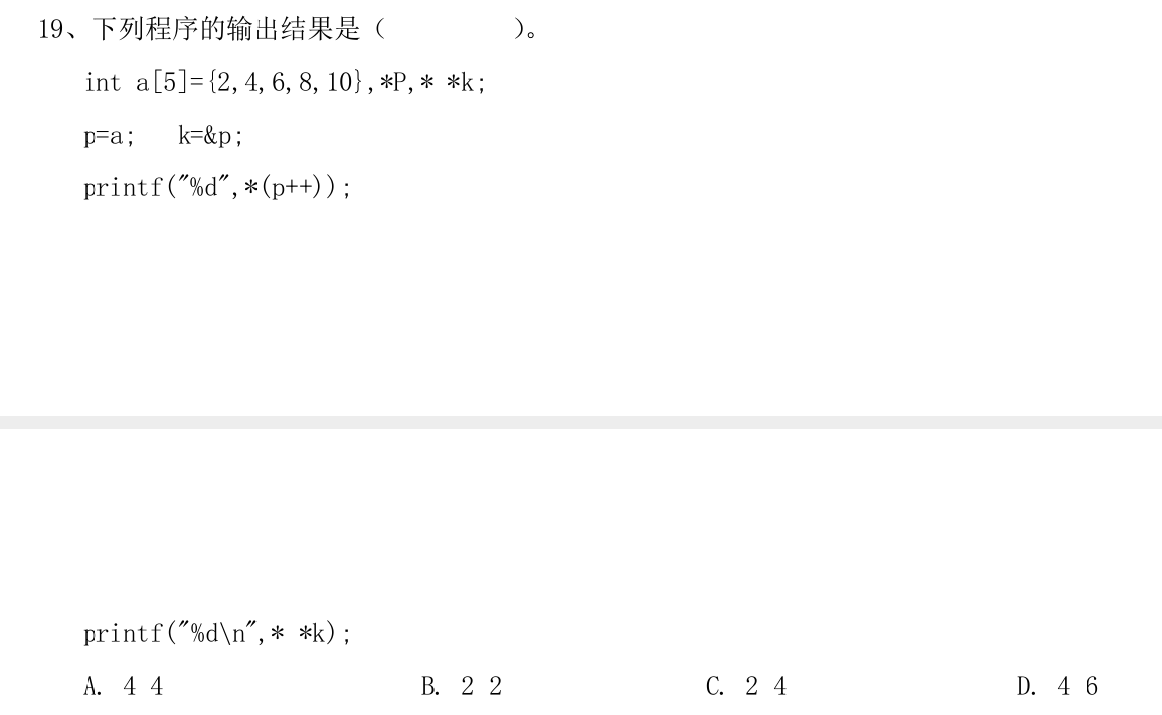
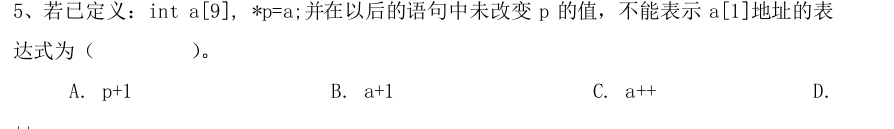






C语言函数可以嵌套调用，但不可以嵌套定义

但结构体可以嵌套定义，且包括自身嵌套和相互嵌套

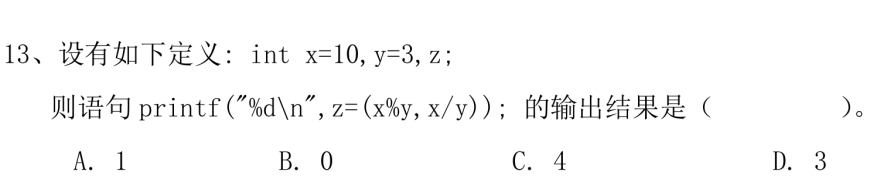


\*（p++）、\*（++p）与\*p++、\*++p

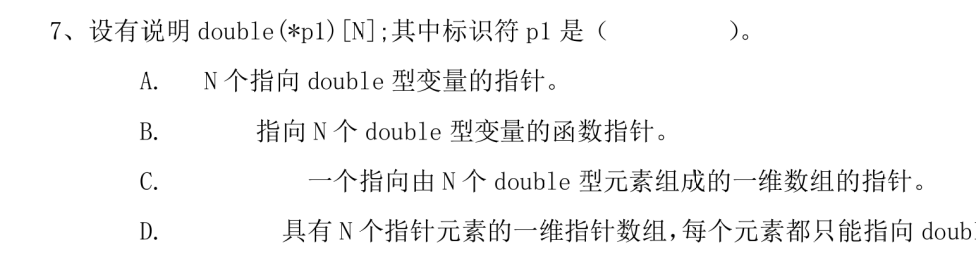
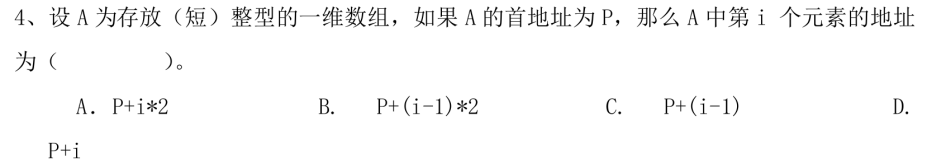
有什么

优先级

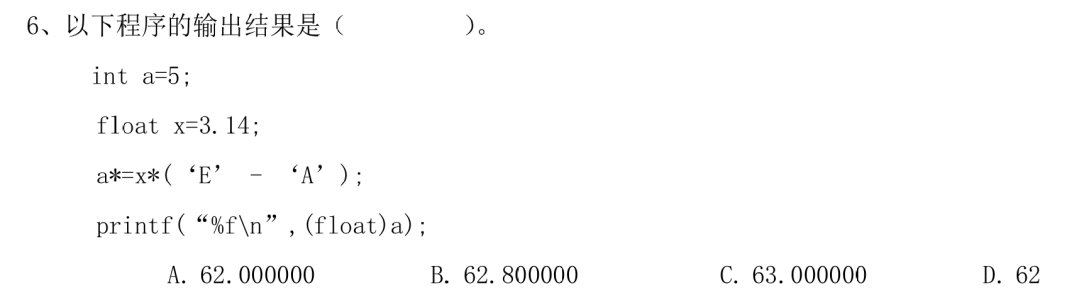
（逗号表达式）取最右



这里说A的首地址是p即p是A所占第一个字节的地址，故此题问的是字节地址。If题目说\*p=A，则p是指数组A的首元地址，则第i个元素地址应为p+i-1

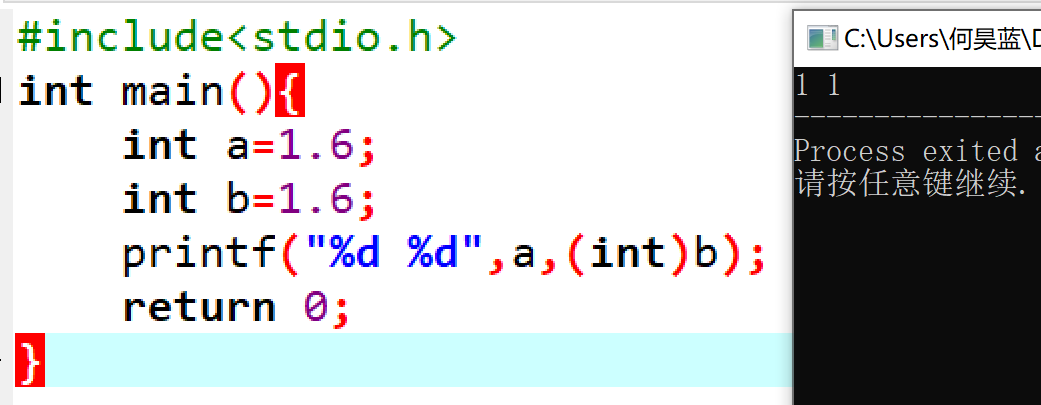


（\*p）[常量表达式]（必须要有）



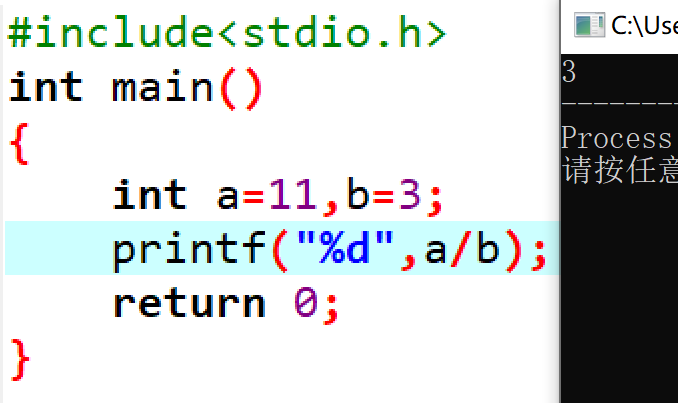
**为什么不是自动转换为float型？**

右侧结果为float，赋值给a时转为int

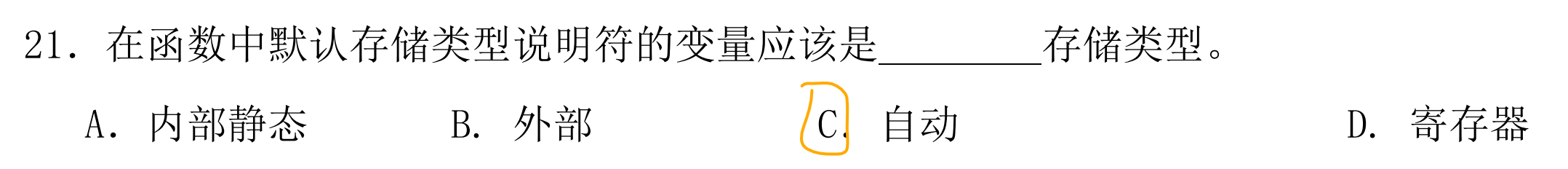
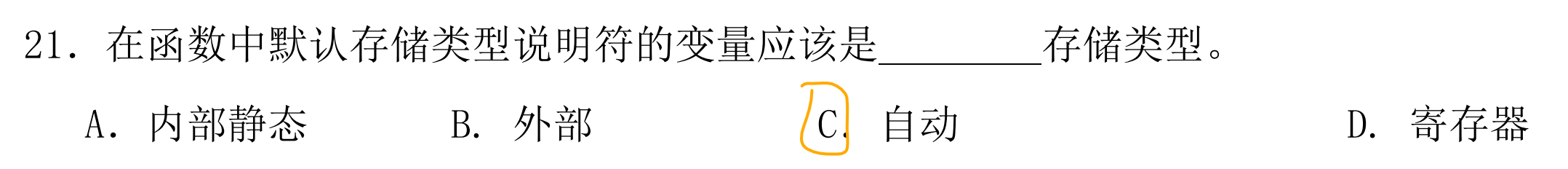


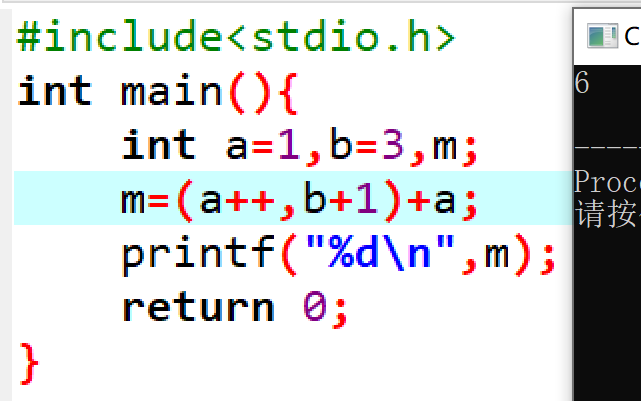
取整丢小数的情况：（都是舍弃小数部分）

１、int a =1.6；    ２、(int)a；

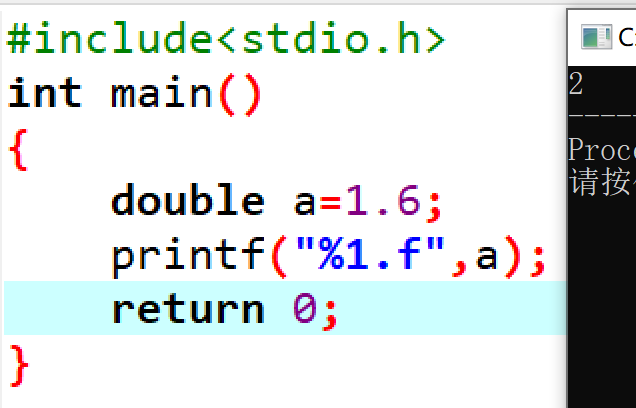


两整数相除也是直接舍弃小数部分

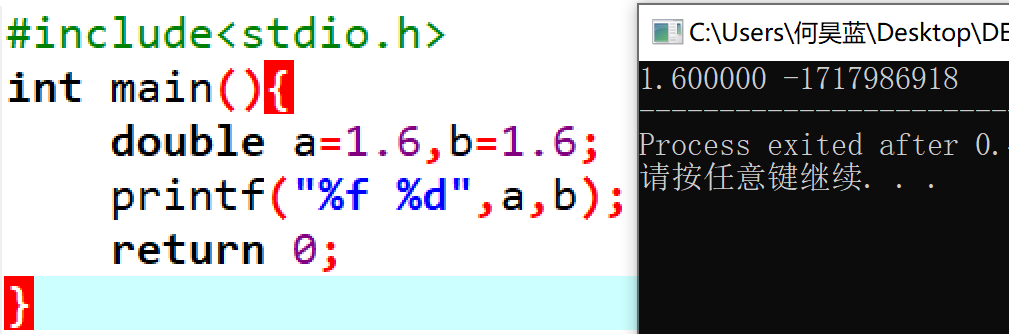




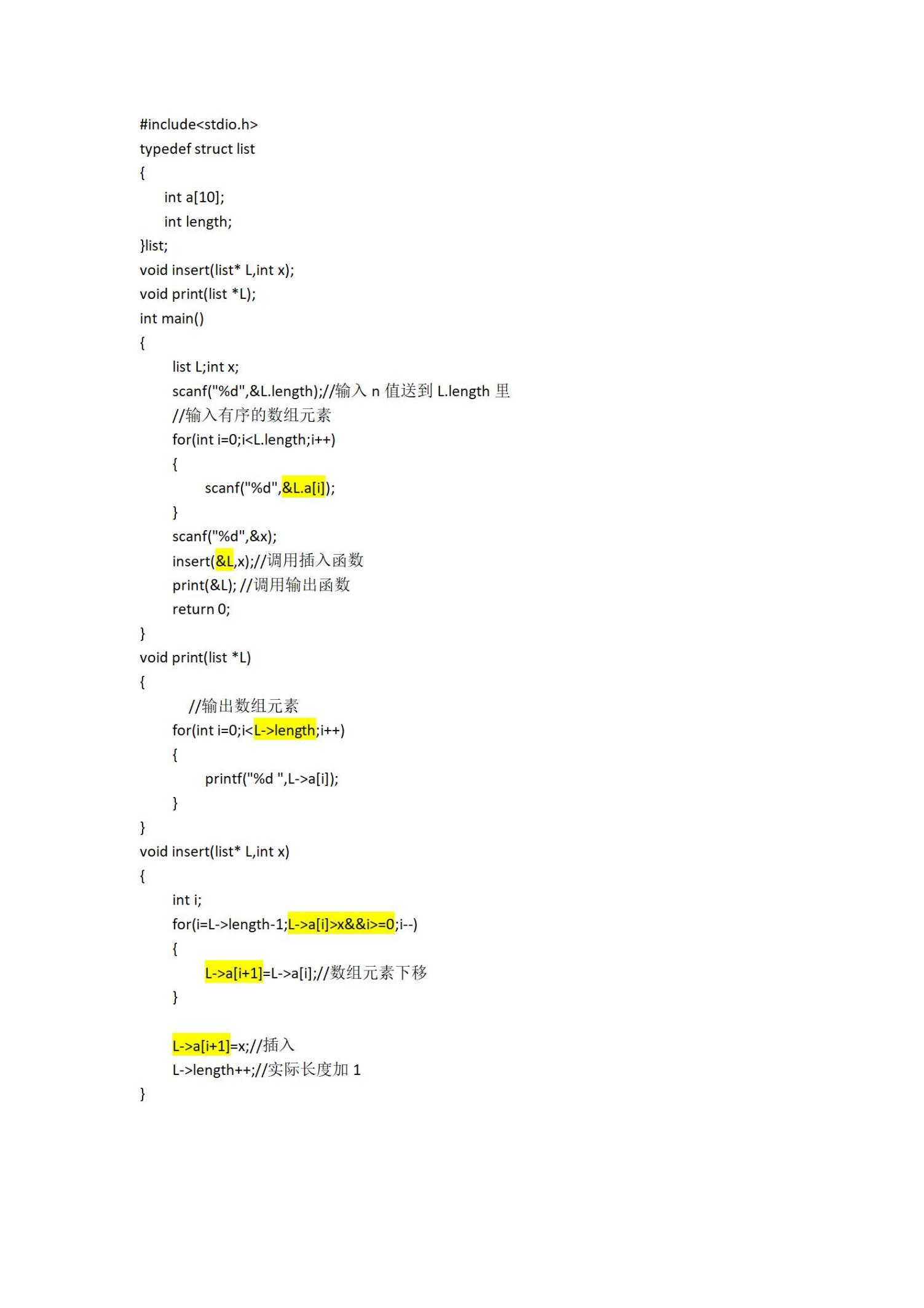
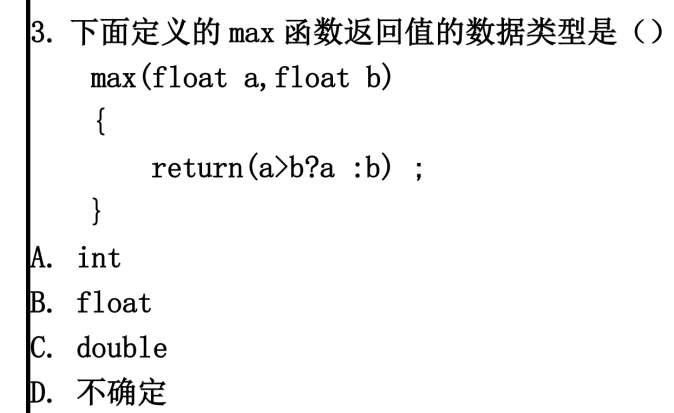
关于括号内自加自减情况



只有此情况会四舍五入



关于double型不同格式（%f、%d）输出情况



选A

Free(大写)可以用作标识符吗？**可以**

形参为list\*L相当于定义变量List\*p

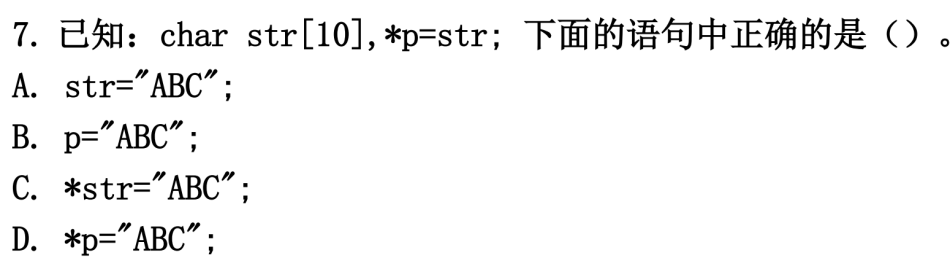
实参为&L

相当于\*p=&L

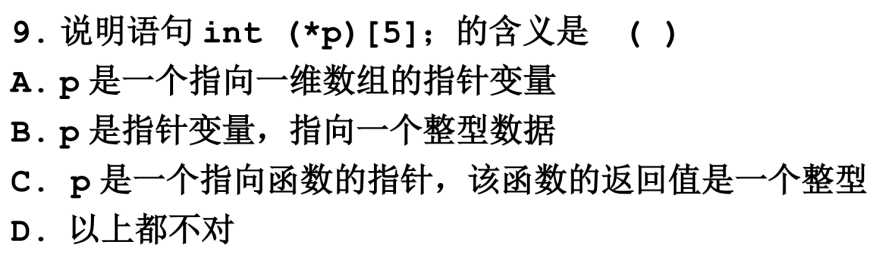
未定义类型的函数，默认为int



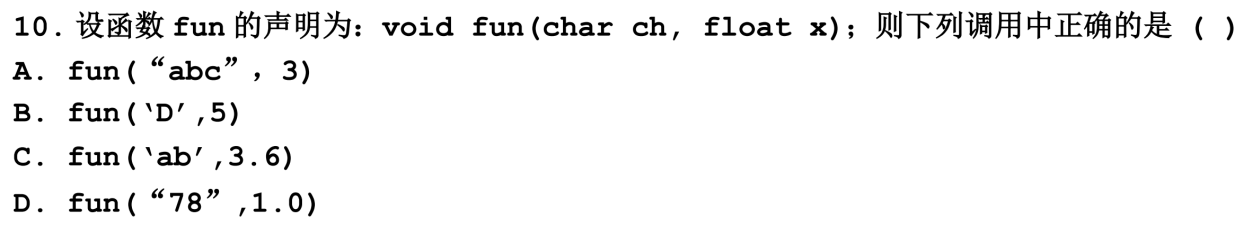
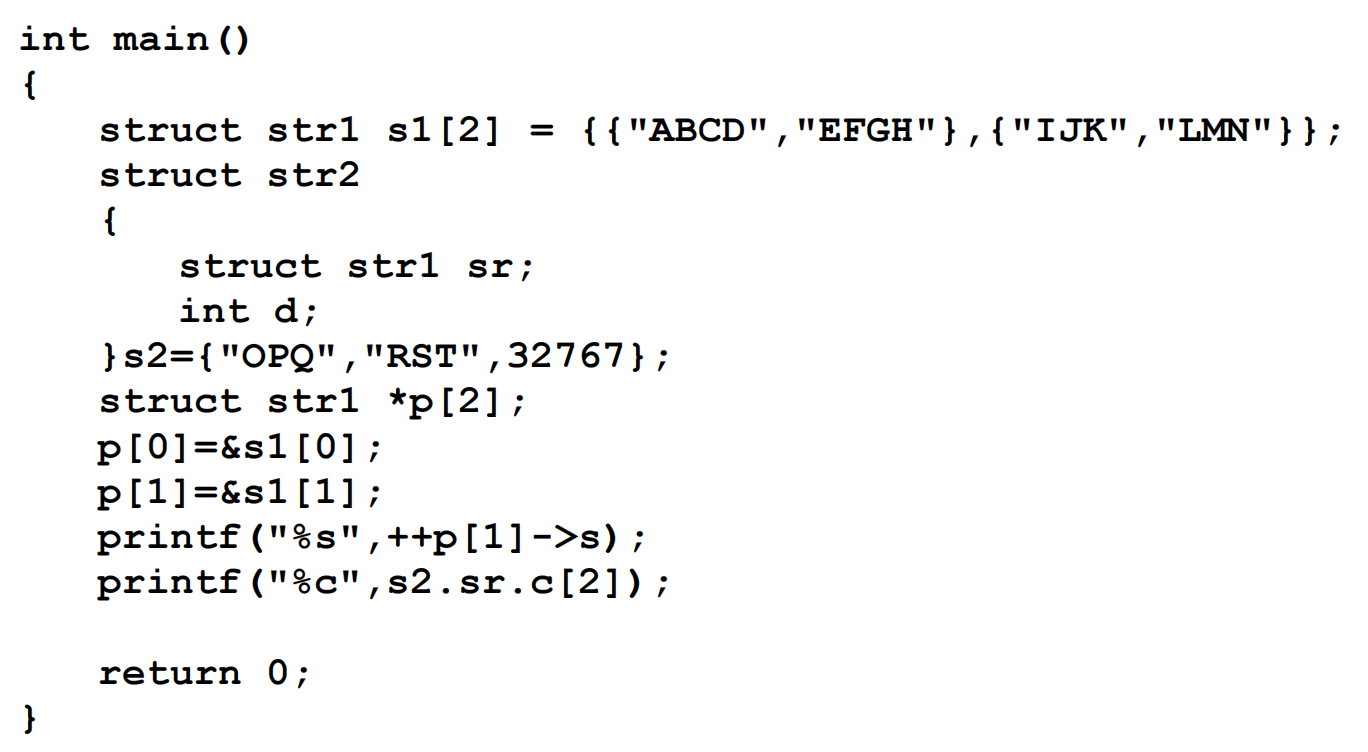
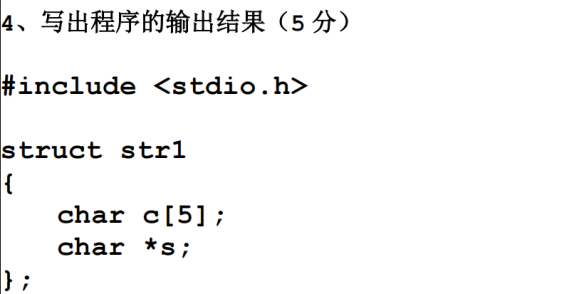
选B



选B（不选D）



行指针变量：选A



选B

答案：MNQ

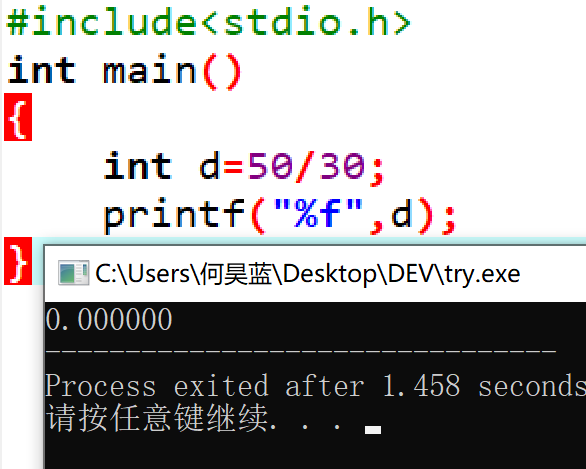
**Strcmp如果第一个字符串在第二个字符串前面，则返回负数；如果两个字符串相同则返回0；第一个字符串在第二个字符串后面（即前者ascii码大），返回正数。**

**注意，不同的系统返回值有可能不一样，取决于系统的实现。有的系统分别返回-1,0,1。在另外的一些系统中，有可能值返回ASCII的差值。**

**#ascii码小写字母在大写字母后面**



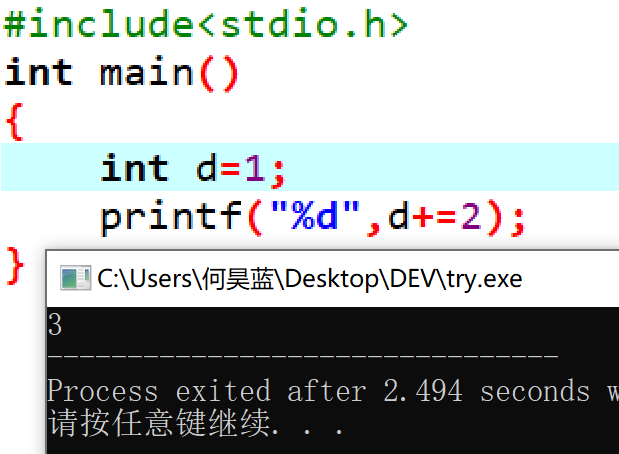
switch中defaut语句：



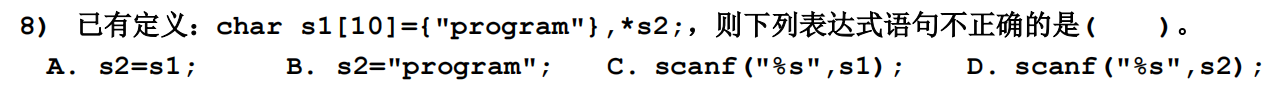
“%f”输出整型数据时，输出结果均为0.000000（默认六位小数）

输出31.0时正常输出31.000000

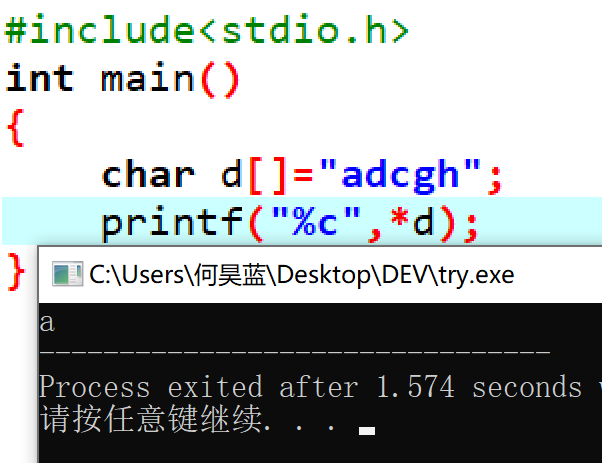




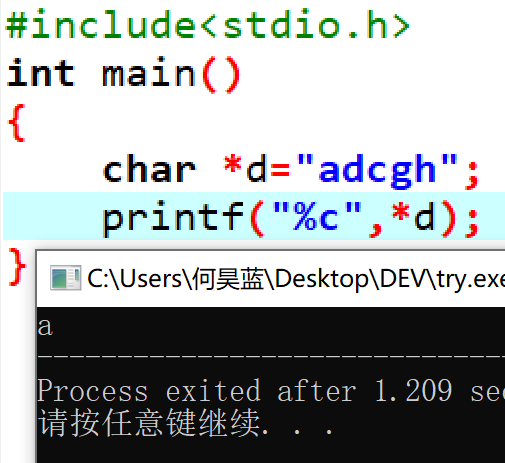
**这样的输出是可以的：**



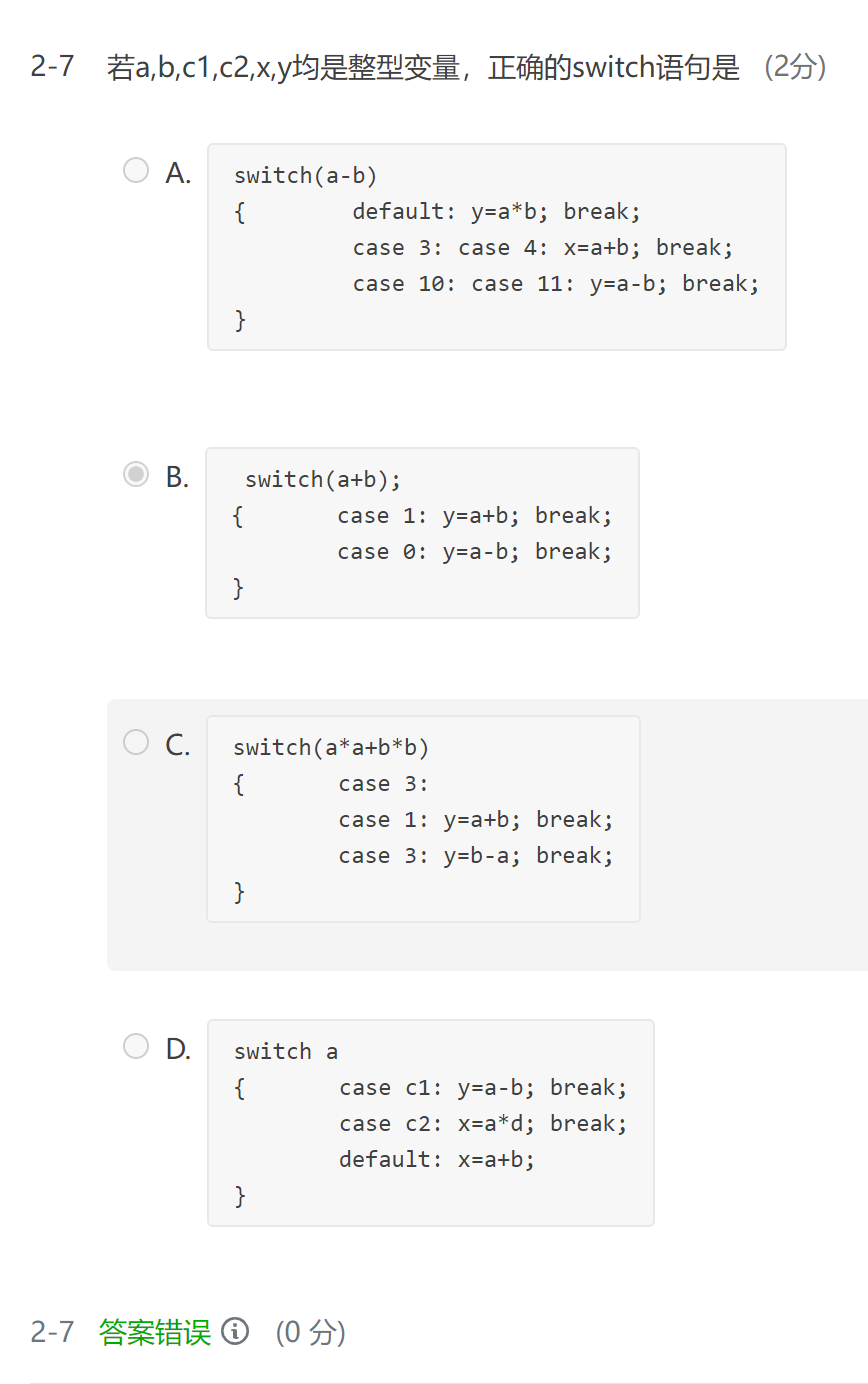
S2未被初始化

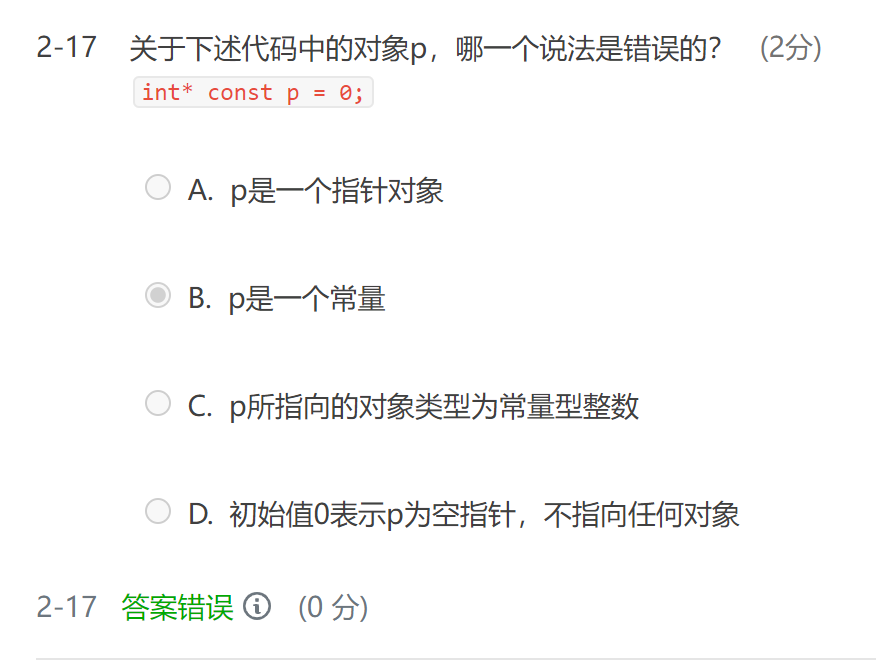


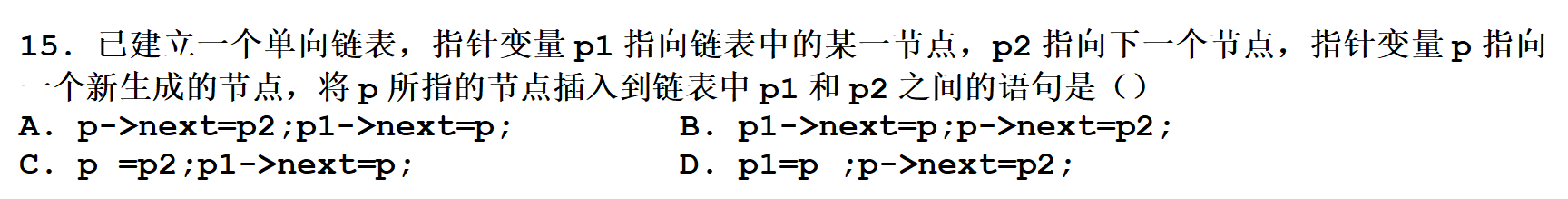
**对比**











A、B

C语言都是有符号数吗？四个字节

所以最小int是-2^31，最大是2^31-1

1. 一个C程序的执行总是从该程序的main函数开始，在main函数最后结束。**×**
2. 函数说明指的是在程序中设定一个函数模块。**×（函数定义）**
3. 在C程序中，若对函数类型未加说明，则函数的隐含类型为void。**×(int)**
4. 形参可以是常量、变量或表达式**×形参只能是变量，实参可以是常量、变量或表达式**
5. 定义函数时，形参的类型说明可以放在函数体内。**×函数返回值在寄存器中, 没有地址, 不能作为形参，但可以作为实参**
6. 定义函数时，形参的类型说明可以放在函数体内。**×**
7. 函数的隐含存储类型是extern。√（外部类型允许其他文件中的函数调用，如果一个函数是不会被其它文件调用的，那么这个函数应该被声明成static的）
8. 全局变量可以被任何一个函数中的任何一个表达式使用。**×位置先后**
9. 函数原型为“void sort(float a[],int n)”，调用该函数时，形参数组a被创建，实参数组各元素的值被复制到a数组各元素中。**×数组首地址传给形参**
10. static char c[]={'a' ,' ','b','o','o','k','\0'};与 static char c[]={'a' ,' ' ,'b','o','o','k'}; 等价。**×**
11. 若有说明：int a[3][4]={0}，则只有元素a[0][0]可得到初值0。**×**
12. 使用带参数的宏时，参数的类型应与宏定义时的一致。**×**
13. 若有宏定义：#define PI 3.14i5926，则系统预编译时会报错。**√（无法识别）**
14. 若有定义：int a[5]={10,20,30,40,50},\*p=a,\*s=a+2;，表达式s-p的值是2。**√**
15. 位运算赋值有6个，但位运算组成复合的赋值运算符只有5个。**√**
16. 在程序中定义了一个结构体类型，将为此类型分配存储空间。**×**
17. feof函数只能对二进制文件进行判定。**×**
18. fp已正确定义并指向某文件，当未遇到该文件结束标志时函数feof(fp)的值为**×**
19. 函数f可以用f(f(x))形式调用，这种调用形式是递归调用**×**
20. C语言可移植性强√
21. 设有定义语句char s[4]="abc";，则表达式s[3]的值是（ B ）。

A. 未知字符 B. 0 C. '0' D. 'c'

1. 设变量定义为 char format[ ]=“s=%d\n”; 则数组format元素的个数为（ 6）。字符串里%d是两个字符
2. 宏定义为“#define div(x，y) x/y；”若语句“printf("div(x，y)=%d\n"，div(x+3，y－3))；”作宏替换后为（ C ）。双引号内宏定义不替换

A．printf("x/y=%d\n"，(x+3)/(y－3))； B．printf("x/y=%d\n"，x+3/y－3)；

C．printf("div(x，y)=%d\n"，x+3/y－3；)； D．printf("x/y=%d\n"，x+3/y－3；)；

1. 程序中调用了库函数exit，必须包含头文件（ B ）。

A．string.h B．stdlib.h C．ctype.h D．math.h

1. 对于以下变量定义，（ D ）是正确的赋值。

int \*p[3],a[3]; **指针数组**

1. p=a B.\*p=a[0] C.p=&a[0]  **D.p[0]=&a[0]**

**一般指针变量的赋初值**

**Way1：int I, \*p;**

**p=&I;**

**Way2：int I,\*p=&I;**

1. 有定义：char \*p[10];，则p是一个（ D ）。

A．函数名   B．指针变量   C．数组元素   D．数组名

1. 指针变量p1、p2类型相同，要使p2、p1指向同一变量，哪一个语句是正确的（ A ）。

A．p2=\*&p1； B．p2=\*\*p1； C．p2=&p1； D．p2=\*p1；

1. 声明语句为“char s[4][15]，\*p1，\*\*p2；int x，\*y；”，下列语句中正确的是（ D ）。
2. p2=s； B．y=\*s； C．\*p2=s； D．y=&x；
3. 定义函数int \*max()的正确含义是（ B ）。

A．返回整型值的函数

B．返回指向整型变量的指针

C．返回指向函数max()的指针

D．上面的叙述都不正确。

1. 若有定义“int(\*ptr)(float)；”，其正确的含义是（ C ）。

A．ptr是指向一维数组的指针变量

B．ptr是指向int型数据的指针变量

C．ptr是指向函数的指针变量，该函数有一个float型参数，返回值是整型

D．ptr是一个函数名，该函数的返回值是指向int型数据的指针

**int （\*P）( )  是函数指针，指向函数的指针，  p 是一个指向函数入口的指针变量， 该函数没有形参  函数的返回值是int 型**

**int  \*p()   是指针函数，返回值为指针的函数     p 是函数的入口地址， 该函数没有形参， 返回值为  int \* 型**

1. int a[12]={1,2,3,4,5,6,7,8,9,10,11,12},\*p[4],i;

p[i]=a+i\*3; ======== p[i]=&a[i\*3];

1. 若变量已正确定义，（ A ）不能使指针p成为空指针。

A．p=EOF     B．p=0          C．p=’\0’       D．p=NULL

1. **注意：**while ( \*t++ = \*s++ ) ; // 注：是单等号
2. 以下程序段输出字符串str，其中循环条件str[i] != '\0' 还可以写成（ A ）。

char str[80];

for(i = 0; str[i] != '\0'; i++)

putchar(str[i]);

A．str[i] != NULL B．i < 80

C．!(str[i] = '\0') （双等号） D．i <= 80

1. 设变量定义为：int n, \*p; char ch, \*q, \*g;，则语句（ p = &n; q = &ch; g = p; ）是不正确的。（指针类型与所指量类型应一致）

*指针可以指向常量，指向常量的指针（！=指针常量）意思是指针所指的对象是一个常量*

1. 设有定义语句

int x[6]={2,4,6,8,5,7},\*p=x,i;

要求依次输出x数组6个元素中的值，不能完成此操作的语句是（ D ）。

A．for(i=0;i<6;i++) printf("%2d",\*(p++));

B．for(i=0;i<6;i++) printf("%2d",\*(p+i));

C．for(i=0;i<6;i++) printf("%2d",\*p++);（\*p++ 使指针p加1）

D．for(i=0;i<6;i++) printf("%2d",(\*p)++);

1. 若结构体定义如下，则下面正确的项是（ C ）

struct time

{ int hour,minute,second; } t;

A．time.hour=5; B．t={15,26,7};

C．struct time b={15,26,7}; D．struct time a.second=7;

1. 若a是整型变量，表达式 ~(a ^ ~a ) 等价于（ C ）。

A．~a           B．1           C．0              D．2

1. 若有以下定义语句：

struct student

{ int num; char name[9];

}stu[2]={1, "zhangsan",2, "lisi"};

则以下能输出字符串“lisi”的语句是（ D ）。

A．printf("%s"，stu[0].name); B．printf("%s"，&stu[1].name);

C．printf("%s"，stu[1].name[0]); D．printf("%s"，&stu[1].name[0]);

1. 对于以下结构定义:

struct

{ int len;

    char \*str;

 } \*p;

 则\*p->str++中的++加在（ A ）。

 A．指针str上   B．指针p上   C．str所指的内容上   D．以上均不是

1. 对于以下的变量定义，表达式（ B ）不符合C语言语法。

考的其实就是运算符的优先级  
优先级最高的是小括号、中括号、点还有->箭头  
.是结构体变量取成员符->是结构体指针取成员符  
A)p->pk p是指针，用->正确  
B)\*p.pk .优先于\*，那就是先对p.pk取内容  
C)\*p->pk ->优先于\*，一样的，不解释了  
D)\*x.pk 同上，不解释，正确

**间接访问运算符优先级低于自增运算（++/--）但高于+=**

struct node {

   int len;

   char \*pk;

 } x = {2, "right"}, \*p = &x;

1. p->p（x->pk错x.pk或(\*x)->pk正确）

B．\*p.pk         C．\*p->pk         D．\*x.pk

1. 对于以下的变量定义，表达式（ D ）是不正确的。

 struct node {

 int j, k;

 } x, \*p=&x;

A．p->k=2  B．(\*p).k=2  C．x.k=2  D．x->k=2

1. 若a是整型变量，表达式 ~(a ^ ~a ) 等价于（ C ）。
2. ~a           B．1           C．0              D．2

异或也叫半加运算，其运算法则相当于不带进位的二进制加法：二进制下用1表示真，0表示假，则异或的运算法则为：相同为0，异为1

优先级：移位>关系运算符 > & > ^ > | > 逻辑运算

1. 执行语句“a=3; printf("%d\n",a&3);”后的输出结果为（ C ）。
2. 1 B． 0 C． 3 D． 11

“&”是按位与，当运算数都为1时结果为1

1. 不能正确打开文件时，打开文件函数fopen的返回值是（ A ）。

A．NULL    B．EOF      C．非0      D．1

1. 定义 FILE \*fp;则文件指针fp指向的是（ D ）。

A.文件在磁盘上的读写位置 B.文件在缓冲区上的读写位置

C.整个磁盘文件 D.文件类型结构

1. 以“只读”方式打开文本文件c:\xy.txt，下列语句中哪一个是正确的（ B ）。

A．fp=fopen("c:\\xy.txt","a"); B fp=fopen("c:\\xy.txt","r"); 字符串中“\\”转义字符表示\

C． fp=fopen("c:\\xy.txt","wb"); D． fp=fopen("c:\xy.txt","r");

1. 对于int a，要使((1<<2>>1)|a)==a;  则a 可以是（ D ）。

A．2      B．6        C．10     D．2,6,10均可

1. 关于关闭文件函数，下列说法中正确的是（ D ）。

A．关闭文件函数没有返回值

B．关闭文件函数的函数名是fclosefile

C．若在程序中打开了某个文件，程序结束前没有关闭这个文件，则会出现语法错误

D．关闭文件函数的参数是指向已经打开的文件的文件型指针变量名（文件结构名指针变量）

1. 设有下列数据定义语句：char s[]="ccc.txt",\*p=s;要打开名为"ccc.txt"的文件，下列给出的打开文件函数的第一个实际参数写法中，错误的是（ B ）。

A．p      B．\*p      C．s      D．"ccc.txt"

int read\_data(ST\* students,char\* file\_name)

{

FILE\*fp;

fp=fopen(~~\*file\_name~~,"r");

int i=0;

while(!feof(fp))

{

fscanf(fp,"%s%s%c%lf%lf%lf\n",students[i].xh,students[i].name,&students[i].sex,&students[i].score[0],&students[i].score[1],&students[i].score[2]);可以

i++;

}

return i;

}

int read\_data(struct Student\* students，char\* file\_name)

{

FILE \*fp,i=0;

fp=fopen(file\_name,"r");

while(fp)

{

fscanf(fp,"%s%s%c%lf%lf%lf\n",(students+i)->xh,(students+i)->name,&(students+i)->sex,&(students+i)->score[%0],&(students+i)->score[1],&(students+i)->score[2]);

(students+i)->total=(students+i)->score[0]+(students+i)->score[1]+(students+i)->score[2];

i++;

}

fclose(fp);

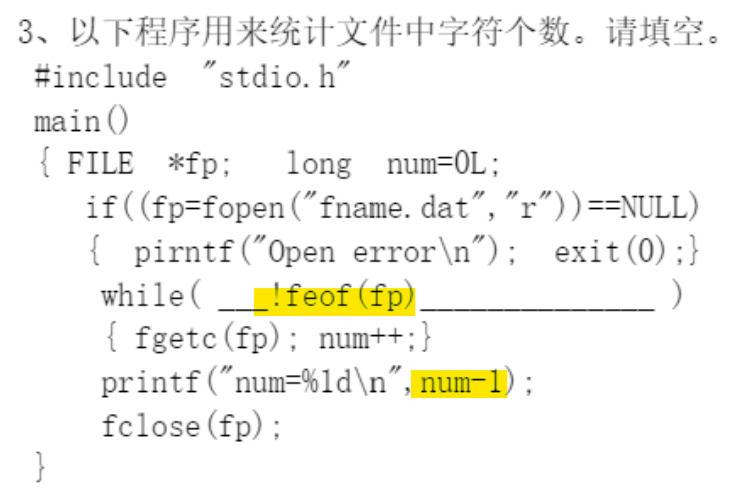
return i;

}

区别字符串和字符数组：

字符串：str[]=”abcd”——系统会自动在最后加’\0’

字符数组：str[]={‘a’,’b’,’c’,’d’}——系统不会自动在最后加’\0’，故不能用strlen



循环以feof为结束条件时

注意统计个数时要减1

