**C语言第1、2章练习题**

**一、选择题**

1. 下列运算符中，（）结合性从左到右。

A. 三目 B. 赋值 C. 比较 D. 单目

1. 下列for循环的次数为（）：for ( i=0，x=0；! x && i<=5；i++)

A. 5 B. 6 C. 1 D. 无限

1. 下述关于循环体的描述中，（）是错误的。
2. 循环体中可以出现break语句和continue语句；B．循环体中还可以出现循环语句；
3. 循环体中不能出现goto语句； D.循环体中可以出现开关语句。

4．下列变量名中，（）是合法的。

A．CHINA； B．student-num； C．double D．A+b

5．有以下程序段： int n=0,p;　　do {scanf(”%d”, &p);n++;} while(p!=12345&&n<3);  
此处do-while循环的结束条件是（）。  
 A．p的值不等于12345并且n的值小于3 B．p的值等于12345并且n的值大于等于3  
 C．p的值不等于12345或者n的值小于3 D．p的值等于12345或者n的值大于等于3

6．若有定义：int a=8，b=5，C；，执行语句C=a/b+0.4;后，c的值为（）

A．1.4 B．1 C．2.0 D．2

7． 以下程序中，while循环的循环次数是（）

**void main( ){**

**int i=0;**

**while(i<10){**

**if(i<1) continue;**

**if(i==5) break;**

**i++; } .....**

A．1 B．10 C．6 D．死循环，不能确定次数

8．下列while循环的执行次数是（）while(i=0) i--;

A．0 B．1 C．5 D．死循环

9．以下说法中正确的是：（）  
A.C语言程序总是从第一个的函数开始执行；  
B.在C语言程序中,要调用的函数必须在main()函数中定义；  
C.C语言程序总是从main()函数开始执行；  
D.C语言程序中的main()函数必须放在程序的开始部分。

10．以下程序的输出结果是：（）

**void main() {  
 int x=10,y=10;   
 printf("%d %d\n",x - -, - - y);   
}**   
 A. 10 10 B.9 9 C. 9 10 D. 10 9

11．**int a=1,b=3;**,则下列表达式的结果为“真”的是：（）  
A. **a>=2||!b&&b<4** B. **b-a&&!a||a-b&&a/b**  
C. **a-!a%b&&!b-!b%b** D.**a+b&&a-b&&b-3&&a||b**

12．请选出合法的C语言赋值语句（）  
A.a=b=58 B.i++; C.a=58,b=58 D.k=int(a+b);

13．若有以下定义和语句：（）  
**char c1='b',c2='e';   
printf("%d,%c\n",c2-c1,c2-'a'+'A');**   
则输出结果是:   
 A. 2,M B. 3,E C. 2,E D. 输出项与对应的格式控制不一致,输出结果不确定。

14．运行以下程序段后变量a的值是：（）

**int a=0, b=3;  
switch(a+b&&a&&b++&&a++)  
{  
 case 0:a++;  
 case 1:b++;  
 case 2:a+=b; break;  
 case 3:a-=b;  
 default:a;  
}**

(A)-3 (B)1 (C)5 (D)6

15.在以下一组运算符中,优先级最高的运算符是：（）  
 (A)<= (B)= (C)% (D)&&

16.设 int a=12，则执行完语句 a+=a-=a\*a后，a的值是（）  
(A)552 (B)264 (C)144 (D)-264

17.C语言中基本数据类型包括 （）  
(A)整型、实型、逻辑型 (B)整型、实型、字符型   
(C)整型、字符型、逻辑型 (D)整型、实型、逻辑型、字符型

18.执行下面程序片段的结果是（）  
**int x=23;   
do{**

**printf("%2d",x - -);**

**}while(!x);**  
(A)打印出321 (B)打印出23 (C)不打印任何内容 (D) 陷入死循环

19.若k是int型变量,且有下面的程序片段：（）  
**k = -3;  
if(k<=0) printf("####")  
else printf("&&&&");**上面程序片段的输出结果是  
 (A)#### (B)&&&& (C)####&&&& (D)有语法错误,无输出结果。

20.为表示关系x≥y≥z,应使用C语言表达式 （）  
(A)(x>=y)&&(y>=z) (B)(x>=y)AND(y>=z) (C)(x>=y>=z) (D)(x>=y)&(y>=z)

21. 若c为char类型变量，能正确判断出c为小写字母的表达式是（）

A) 'a'<=c<='z' B) (c>='a') || (c<='z') C) c>='a' || c<='z' D) c<='z' && c>='a'

22．下面程序执行后输出sum的值是（）

A)15 B) 14 C) 0 D) 不确定

**void main()**

**{ int i,sum;**

**for(i=0;i<6;i++) sum+=i;**

**printf("%d\n",sum);**

**}**

23.以下叙述正确的是：（）

A) 可以把include和if定义为用户标识符

B) 可以把include定义为用户标识符，但不能把if定义为用户标识符

C) 可以把if定义为用户标识符，但不能把include定义为用户标识符

D) include和if都不能定义为用户标识符

**二、 判断下列描述的正确性，对者划√，错者划╳。**

1. switch语句中必须有break语句，否则无法退出switch语句。
2. C中标识符内的大小写字母是没有区别的。
3. 隐含的类型转换都是保值映射，显式的类型转换都是非保值映射。
4. 运算符的优先级和结合性可以确定表达式的计算顺序。
5. while循环语句的循环体至少执行一次。
6. 开关语句不可以嵌套，在开关语句的语句序列中不能再有开关语句。
7. C语言在编译时不检查语法；
8. 在C语言中，**”A”**是错误的char类型的常数；
9. 字符(char)型数据在微机内存中的存储形式是ASCII码；
10. C语言的源程序不必通过编译就可以直接运行；
11. **“Int”**是合法的C语言关键字；

**三、写出下列程序的运行结果。**

2.

void main(){

int i =1, sum=0;

while (i<=5){

sum+=i; i++;}

printf(“sum=%d”, sum);

}

1.

void main(){

int a=4,b=5,c=0,d;

d=!a&&!b||!c;

printf(“%d”, d);

}

3.

void main(){

int a=0,i;

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

switch(i){

case 0:

case 3:a+=2;

case 1:

case 2:a+=3;

default:a+=5;

}

}

printf(“%d”, a);

}

4.

void main()  
{int a=15,b=21,m=0;  
switch(a%3)  
 {case 0:m++;break;  
 case 1:m++;  
 switch(b%2)  
 {default:m++;  
 case 0:m++;break;  
 }

};  
printf(“%d”，m);}

6.

void main(){

int a=5,b=4,c=3,d;

d=(a>b>c);

printf("%d\n",d);

}

**5．从键盘输入1325。**

void main(){

int n1,n2;

scanf("%d",&n2);

while(n2!=0) {

n1=n2%10;

n2=n2/10;

printf("%d",n1); }

}

8.

void main() {

int i;

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

if(i%2) printf("\*");

else continue;

printf("#");}

printf("$\n");

}

7.

void main(){

int s,i;

for(s=0,i=1;i<3;i++,s+=i);

printf("%d\n",s);}

10.

//输入m的值为18

#include <stdio.h>

void main( ){

int m,i;

scanf("%d", &m);

for(i=1;i<m;i++)

if(m%i==0) printf("%3d",i);

}

9.

void main ( ){

int a,b;

for (a=1,b=1 ; a<=100 ; a++) {

if (b>=20) break;

if (b%3==1) { b+=3 ; continue ; }

b-=5; }

printf("%d\n",a);

}

**四、完成程序。**

**5.求某数的泰勒（台劳）级数的前n+1项之和。x的泰勒级数：**

**** void main ( ){

int i,n; float x; float t=1.0,sum=1.0;

scanf(“%f, %d”, &x, &n);

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

t\*= **（9）**;

**（10）**= t; }

printf(“%f”, sum);

}

**1. 程序的功能是:分别计算1到10之间奇数之和及偶数之和**

void main(){

int a,b,c,i; a = c = 0;   
for(i=0;i<=10;i+=2) {

a += i;  
 b =**（1）;** c += b; }  
printf("偶数之和=%d\n",a);   
printf("奇数之和=%d\n",**（2）**); }

**6. 得到一个输入数字的反转数，然后一次输出这个整数。**

void main()

{int n, right\_digit, newnum = 0;

printf("Enter the number: ");

scanf("%d",&n);

printf("reverse order is ");

do { right\_digit = n % 10;

//生成所输入数字的反转数

newnum=newnum\*10+right\_digit;

n =**（11）** ;

} while (n != 0);

printf("%d\n",**（12）**);

}

**2. 输出100以内能被3整除且个位数为6的所有整数。**

void main() {

int i, j;   
 for(i=0; **（3）**; i++) {  
 j=i\*10+6;   
 if(**（4）**) continue;   
 printf("%d",j); }   
}

**3**. **判断输入的正整数是否为素数。**

#include <math.h>  
void main(){

int m，i，k;  
 scanf("%d"，&m);  
 k=sqrt(m);  
 for (i=2;i<=k;i++)  
 if(**（5）**) break;  
 if(**（6）**) printf("%d 是素数\n"，m)

else printf("%d 不是素数\n"，m); }

**7．猴子吃桃问题。猴子第一天摘下若干个桃子，当即吃了一半多一个。第二天早上又将剩下的桃子吃掉一半多一个。以后每天早上都吃掉前一天剩下的一半多一个。到*第5天早上吃时*，只剩下一个桃子。求第一天共摘了多少个桃子。**

void main(){

int day, x1,x2; //day是天数， x1和x2表示前一天和当天剩下的桃子数

day=4; x2=1;//第4天剩下的桃子数

while(**（13）**){

x1=(x2+1)\*2;// 当天剩桃子数：x2=x1/2-1

x2=**（14）**;

**（15）**;

}

printf(“桃子总数=%d”,x1);}

**4. 输入学号，并输出其中能被7或9整除的学号，当学号输入0值时结束循环。**

void main(){

int num;

do{ scanf (“%d”, &num);

if(**（7）**)

printf(“%d ”, num);

} while (**（8）**);

}

**五、程序设计题**

**2. 输入一行字符，分别统计出其中英文字母、空格、数字和其它字符的个数。**

void main(){

char ch;

int letter=0,space=0, digit=0,other=0;

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

if((ch>='a' && ch<='z') ||(ch>='A' && ch<='Z'))

letter++;

else if (ch>='0' && ch<='9') digit++;

else if(ch==' ') space++;

else other++;

}

printf("%d,%d,%d,%d",letter,digit,space,other);

}

**1.求1-5的阶乘之和。**

n!=(n-1)! \* n sum:和。初值为0。

factorial:积。初值为1。

#include <stdio.h>

void main()

{int i, factorial=1, sum=0;

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

factorial\*=i;

sum+=factorial;

}

printf(“%d”,sum);

}

**3.求Sn=a+aa+aaa+…+aa…a 的值。其中a是一个数字。n由键盘输**入。

Sn=term1+term2+term3+…+termn

term1 =a term2 = term1\*10+a

termi+1 = termi\*10+a

void main()

{int n,a,,i; float f,term;

scanf("%d,%d", &a, &n);

term=a; sum=term;

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

//termi+1 = termi\*10+a

term=term\*10+a;

sum+=term;

}

printf("sum=%f",sum); **}**

**4. 求前20项之和： …… (选做)**

找规律：



void main(){

float sum=0; int a=1,b=2, t;

for(int i=1; i<=20; i++) {

sum+=(float)b/a; //sum+=bi/ai

t=a+b; //t=ai+bi

a=b; //ai+1=bi

b=t; //bi+1=ai+bi

}

printf(“%f”,sum);

}



**6. 输入两个正整数m和n, 用辗转相除法求最大公约数和最小公倍数。**（**选做**）

void main(){

int a,b,m,n,temp;

printf("Please input the two number:\n");

scanf("%d,%d",&m,&n);

if(m<n)//将两个数中较大的放在m中

{temp=m; m=n; n=temp;}

a=m;b=n;

while(b!=0) /\*a中放的是最大公约数\*/

{temp=a%b; a=b; b=temp; }

printf("最大公约数是：%d\n",a);

printf("最小公倍数是：%d\n",m\*n/a);

}

**5. 打印图案**。（**选做**）

void main(){

int i,j,k;

for(i=0;i<=3;i++){

for(j=0;j<=2-i;j++) printf(" ");

for(k=0;k<=2\*i;k++) printf("\*");

printf("\n");

}

for(i=0;i<=2;i++) {

for(j=0;j<=i;j++) printf(" ");

for(k=0;k<=4-2\*i;k++) printf("\*");

printf("\n"); }

}

7. 设计具有以下功能的程序：从键盘上输入若干个非负整数，统计并输出最大数、最小数和平均数，当输入负数时结束输入。(自己完成，并上机验证)

**一、选择题**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **C** | **B** | **C** | **A** | **D** | **B** | **D** | **A** | **C** | **D** |
| **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **D** | **B** | **B** | **C** | **C** | **D** | **B** | **B** | **D** | **A** |
| **21** | **22** | **23** |  |  |  |  |  |  |  |
| **D** | **D** | **D** |  |  |  |  |  |  |  |

**二、判断题**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **F** | **F** | **F** | **T** | **F** | **F** | **F** | **T** | **T** | **F** | **F** |

**三、读程序**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **1** | **sum=15** | **31** | **1** | **5231** | **0** | **5** | **\*#\*#\*#$** |
| **9** | **10** | **11** | **12** |  |  |  |  |
| **8** | **1 2 3 6 9** |  |  |  |  |  |  |

**四、程序填空**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **（1）** | **（2）** | **（3）** | **（4）** | **（5）** |
| **i+1** | **c** | **i<10** | **j%3** | **m%i==0** |
| **（6）** | **（7）** | **（8）** | **（9）** | **（10）** |
| **i==k+1** | **num%7==0||num%9==0** | **num!=0** | **x/i** | **sum+** |
| **（11）** | **（12）** | **（13）** | **（14）** | **（15）** |
| **n/10** | **newnum** | **day>0** | **x1** | **day--** |

**第3-4章 练习题**

1. **选择题**

**1. 以下错误的字符串赋值或赋初值方式是：**

**（A）char str1[]=”string”, str2[]=”12345678”;strcpy(str2,str1);**

**（B）char str[7]={’s’,’t’,’r’,’i’,’n’,’g’};**

**（C）char str[10] ;str =”string”;**

**（D）char \*str; str=”string”;**

**2. 数组定义中不正确的是 ：**

**（A） int a[2][3]; （B） int b[][3]={0,1,2,3};**

**（C） int c[100][100]={0}; （D） int d[3][]={{1,2},{1,2,3},{1,2,3,4}};**

**3. 以下对一维数组a正确初始化的是：**

**（A）int a[10]=(0,0,0,0,); （B）int a[10]={};**

**（C）int a[10]={0}; （D）int a[10]=(10\*2);**

**4.以下选项中，不能正确赋值的是：   
（A） char s1[10];s1="Ctest"; （B） char s2[]={'C', 't', 'e', 's', 't'};**

**（C） char s3[20]="Ctest"; （D） char \*s4="Ctest\n"**

**5.若有说明：int i, j=2，\*p=&i;，则能完成i=j赋值功能的语句是：   
（A） i=\*p; （B） \*p=\*&j; （C） i=&j; （D） i=\*\*p;**

**6. 下面各语句行中,不能正确进行字符串操作的语句行是：  
(A) char st[10]={"abcde"}; (B) char s[5]={'a','b','c','d','e'};  
(C) char \*s; s="abcde"; (D) char \*s; scanf("%s",s);**

**7.设有如下的程序段,执行上面的程序段后，**\*(ptr+5)**的值为:**char str[ ]="Hello"; char \*ptr; ptr=str; **(A) 'o' (B) '\0' (C)不确定的值 (D) 'o'的地址**

**8. 请选出以下语句的输出结果**printf("%d\n",strlen("\t\"\065\xff\n")); **(A)5 (B)14 (C)8 (D)输出项不合法，无正常输出**

**9.以下程序段给数组所有的元素输入数据，请选择正确答案填入。  
main()  
{ int a[10],i=0;  
 while(i<10) scanf("%d",\_\_\_\_\_\_\_\_ );  
┇}  
 (A)a+(i++) (B)&a[i+1] (C)a+i (D)&a[++i]**

**10.若有以下说明和语句，请选出哪个是对c数组元素的正确引用**int c[10], \*cp; cp=c; **(A)cp+1 (B)\*(cp+3) (C)\*\*(cp+1)+3 (D)\*(\*cp+2)**

**11.设有如下一段程序, 执行下面的程序后, ab的值为：**int \*var, ab; ab = 100; var = &ab; ab = \*var + 10; **(A)120 (B)110 (C)100 (D)90**

**12.若有以下定义：int a[10], \*p=a ；则p+4表示**

**A) 元素a[4]的地址 B) 元素a[4]的值 C) 元素a[5]的地址 D) 元素a[5]的值**

**13. 判断字符串s1是否小于字符串s2，应当使用**

**A) if(s1 < s2) B) if(strcmp(s1,s2)) C) if (strcpy(s1,s2)<0 ) D) if(strcmp(s1,s2)<0)**

**14.在下面的定义语句中，错误的是**

**A)int n=20, a[n]; B) char \*a[3]; C) char s[20]="test"; D) int a[]={1,2};**

**15.对C语言字符数组的描述中错误的是：**

**A) 字符数组可以存放字符串。**

**B) 字符数组的字符串可以整体输入输出。**

**C) 可以在赋值语句中通过赋值运算符对字符数组整体赋值。**

**D) 不可以用关系运算符对字符数组中的字符串进行比较。**

**16.设有以下说明语句：**

**struct stu**

**{ int a; float b; } stutype;**

**则下面的叙述不正确的是：**

**A) struct 是定义结构体类型的关键字。 B) struct stu是用户定义的结构体类型。**

**C) stutype是用户定义的结构体类型名。 D) a 和b都是结构体成员名。**

1. **读程序。**

**5.void main( ){**

**int n[3],i,j,k;   
 for(i=0;i<3;i++)   
 n[i]=0;   
 k=2;   
 for (i=0;i<k;i++)   
 for (j=0;j<k;j++)   
 n[j]=n[i]+1;   
 printf("%d\n", n[1]);   
}**

**void main( ){**

**char \*p="abcd\0efgh\0";**

**printf("%d"，strlen（p）);**

**}**

**2.**

**void main( ) {**

**char ss[][20]={"1234","56789"}, \*p=sss[0];**

**printf("%s", p+20);**

**}**

**6. void main ( ) {**

**int a[5]={15, 23, 4, 67, 98};**

**int i, m, n;**

**m=n=a[0];**

**for (i=1; i<5; i++ )**

**{if (m>a[ i ] ) m=a[ i ];**

**if (n<a[ i ] ) n=a[ i ]; }**

**printf ( “%d , %d”, m, n);**

**}**

**3.**

**void main( ){**

**char b[]="Hello,you";   
 b[5] = b[9];   
 printf("%s \n", b );**

**}**

**4.**

**void main( ){**

**int arr[]={40,30,25,35,12,6,8}, \*p=arr;**

**p++;**

**printf("%d\n",\*(p+4));**

**}**

**7. void main( ) {**

**int x[]={10,20,30,40,50,60,70,80,90,100};**

**int s, i, \*p;**

**s=0; p=x;**

**for(i=1;i<10;i+=2)**

**s+=\*(p+i);**

**printf("sum=%d\n",s)**

**}**

**12.**

**void main ( ) {**

**int i=0; char a[]="aqid",**

**b[ ]="abm", c[10];**

**while (a[i]!='\0' && b[i]!='\0') {**

**if (a[i]>=b[i]) c[i]=a[i]-32 ;**

**else c[i]=b[i]-32 ;**

**i++;}**

**c[i]='\0';**

**puts(c);**

**}**

**8.划线两输出语句分别输出13ff50和4**

**void main( )**

**{ int a[3][4];**

**printf("\n%x", a);**

**printf("\n%d",sizeof(a[0][0]));**

**printf("\n%x", a[0]+2);**

**printf("\n%x", &a[1]+1);**

**printf("\n%x", &a[0][0]+3);**

**}**

**9.**

**void main( ){**

**int i=1,n=3,j,k=3;**

**int a[5]={1,4,5};**

**while(i<=n && k>a[i])**

**i++;**

**for(j=n-1;j>=i;j--)**

**a[j+1]=a[j];**

**a[i]=k;**

**for(i=0;i<=n;i++)**

**printf("%3d",a[i]);**

**}**

**13.**

**#include<string.h>  
void main()  
{char \*p1,\*p2,str[50]=“ABCDEFG”;  
p1=“abcd”; p2=“efgh”;  
strcpy(str+1,p2+1);  
strcpy(str+3,p1+3);  
printf(“%s”,str);  
}**

**14.**

**void main() {**

**char s1[20]="abcd",s2[20]="ABCD";**

**char \*p1=s1,\*p2=s2;**

**printf("%s\n",strcat(p1+2,p2+1));**

**}**

**10.**

**void main(){**

**int a[10]={10,12,13,14,15,16,17,18,19,11};**

**int k,s,i; float ave;**

**for(k=s=i=0;i<10;i++){**

**if(a[i]%2==0) continue;**

**s+=a[i]; k++; }**

**if(k!=0){**

**ave=(float)s/k;**

**printf("%d, %f\n",k,ave); }**

**}**

**15.**

**#include <stdio.h>**

**#include <string.h>**

**void main() {**

**char s1[20]="abcd",s2[20]="ABCD",**

**str[20]="xyz";**

**char \*p1=s1,\*p2=s2;**

**strcat(p1,p2+1);**

**strcpy(str+2,p1);**

**printf("%s\n",str);**

**}**

**11.**

**#include <string.h>**

**void main(){**

**char arr[2][4];**

**strcpy(arr[0],"you");**

**strcpy(arr[1],"me");**

**arr[0][3]='&';**

**printf("%s\n",\*arr);**

**}**

**填空题**

**1.已知字符’A’的ACSII码值为65，以下语句的输出结果是：B\_  
   char ch=’B’;  
   printf("%c\n",ch++);**

**2.** 指针变量的类型用来指定\_该指针变量可以指向的变量\_\_\_的类型；

**3.** 有如下程序段  
**int \*p,a=10,b=1;  
p=&a; a=\*p+b;**   
执行该程序段后，a的值为：\_11\_\_\_；

**4.** 下面程序输出结构体变量sz所占存储单元的字节数。

struct st {char name[20]; float score; } ;

main()

{ struct st sz;

printf("sz size: %d\n", sizeof(struct st/ sz ));

}

1. **编程题目**

**1. 比较两个字符串。（不要用strcmp）**

**void main()**

**{char s1[20],s2[20];**

**int i;int result;**

**gets(s1); gets(s2);**

**i=0;**

**while( (s1[i]= =s2[i])&&(s1[i]!='\0') )**

**i++;**

**result=s1[i]-s2[i];**

**printf("%d\n", result);**

**}**

**2.有一个3×4的矩阵，要求编程序求出其中值**

**最大的那个元素的值，以及其所在的行号和列号。**

**void main()**

**{int i,j,row=0,colum=0,max;**

**int a[3][4]={{1,2,3,4},{9,8,7,6},{-10,10,-5,2}};**

**max=a[0][0];**

**for (i=0;i<=2;i++)**

**for (j=0;j<=3;j++)**

**if (a[i][j]>max)**

**{max=a[i][j]; row=i; colum=j; }**

printf("max=%d,row=%d,colum=%d\n",max,row,colum);

}

**4. 将两个字符串连接起来，不要用strcat。**

**void main()**

**{char s1[20],s2[20]; int i;int j;**

**gets(s1); gets(s2);**

**i=0;**

**while(s1[i]!=‘\0’) /\*找到s1的尾 \*/**

**i++;**

**j=0;**

**while(s2[j]!=‘\0’) /\*逐个拷贝相应字符 \*/**

**s1[i++]=s2[j++];**

**s1[i]='\0';**

**puts(s1);**

**}**

**3. 将字符数组s2中的全部字符拷贝到字符数组s1中。不要用strcpy**

**#include <stdio.h>**

**void main()**

**{char s1[20],s2[20];**

**int i; gets(s2);**

**i=0;**

**while(s2[i]!='\0')**

**{s1[i]=s2[i]; i++;}**

**s1[i]='\0';**

**puts(s1);**

**}**

**5. 编写一个程序, 输入一个整形数组array，将该数组的最后一个元素移到第0个位置，其它元素后移一个位置。在main中将该数组输出。不要定义辅助数组。**

void main(){

int array[10];

int i,t;

for(i=0;i<10;i++)

scanf("%d",&array[i]);

t=array[9];

for(i=9;i>0;i--)

array[i]=array[i-1];

array[0]=t;

for(i=0;i<10;i++)

printf(" %d",array[i]);

}

}

**一、选择题**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **C** | **D** | **C** | **A** | **B** | **D** | **B** | **A** | **A** | **B** |
| **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **B** | **A** | **D** | **A** | **C** | **C** |  |  |  |  |

**二、写出程序运行结果**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **4** | **56789** | Hello | **6** | 3 | **4,98** | **sum=300** | **13ff58**  **13ff70**  **13ff5c** |
| **9** | **10** | **11** | **12** | **13** | **14** | **15** |  |
| **1 3 4 5** | **5, 15.000000** | **you& me** | **AQM** | **Afgd** | **cdBCD** | **xyabcdBCD** |  |

**三、填空题**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **2** | **3** | **4** |
| **B** | 该指针变量可以指向的变量 | 11 | struct st/ sz |

**第5-7章 练习题**

1. **选择题**

**1.有下列程序，执行后由键盘键入2时，s的值为： D**

**#define P (y\*y+3\*y)**

**void main()**

**{int s,y;**

**scanf("%d",&y);**

**s=3\*P+4\*P;**

**printf("s=%d\n",s);}**

**（A） 36 （B）40 （C）52 （D）70**

**2.在C程序中，main( )的位置： C**

**（A）必须作为第一个函数 （B）必须作为最后一个函数**

**（C）可以任意 （D）必须放在它所调用的函数之后**

**3. 以下叙述中正确的是： A**

**（A）构成C程序的基本单位是函数 （B） 可以在一个函数中定义另一个函数**

**（C） main()函数必须放在其它函数之前（D） 所有被调函数一定要在调用之前进行定义**

**4. 在宏定义 #define PI 3.14159中，用宏名PI代替一个 D  
(A)单精度数 (B)双精度数 (C)常量 (D) 字符串**

**5. C语言规定，函数返回值的类型是由 B**

**A) return语句中的表达式类型所决定 B) 由被调用函数的类型所决定**

**C) 由主调函数中的实参数据类型所决定 D) 由被调函数中的形参数据类型所决定**

**6. 调用函数时，实参是一个数组名，则向函数形参传递的是 C**

**A) 数组的长度 B) 数组每个元素的值**

**C) 数组的首地址 D) 数组中每个元素的地址**

**7.设有一函数 f(int b[ ]），在某一主调用函数中有f（a），其中a是一个整型数组且已赋值，则正确的叙述是： C**

**A) a数组与b数组各占用不同的存储空间。 B) 对b数组值的修改不影响a数组的值。**

**C) 对b数组元素值的修改实际上就是修改a数组。 D) 实参与形参的结合是双向传递。**

**8. 以下正确的说法是：在C语言中， A**

**A) 实参变量和与其对应的形参变量各占用独立的存储单元。**

**B) 实参变量和与其对应的形参变量共占用同一个存储单元。**

**C) 当实参变量和对应的形参变量同名时，才占用相同的存储单元。**

**D) 形参变量是虚拟的，不占用存储单元。**

**9.若已定义的函数有返回值，则以下关于该函数调用的叙述中错误的是 D**

**A）函数调用可以作为独立的语句存在 B）函数调用可以作为一个函数的实参**

**C）函数调用可以出现在表达式中 D）函数调用可以作为一个函数的形参**

**10.有以下函数定义：void fun(int n, double x) { …… } C**

**若以下选项中的变量都已正确定义并赋值，则对函数fun的正确调用语句是**

**A）fun(int y,double m); B）k=fun(10,12.5);**

**C）fun(x,n); D）void fun(n,x);**

1. 写出程序运行结果

**2.**

**#define f(x) x\*x  
void main() { int i;  
 i = f(4+4) / f(2+2);  
 printf("i=%d\n",i); }**

**1.**

**#define AA 100**

**void main( )**

**{ printf("AA"); }**

**4.程序输出的最后一个值是**

**void main()**

**{int n = 2, m = 4;  
printf(“%d,”,fun(n, m));  
printf(“%d”,fun(n, m)); }**

**int fun(int a, int b)**

**{static int n; int m = 0;  
n -= a+1; m += b-2;  
return (n+m); }**

**int fun(int n)**

**{static int f=1;**

**f=f\*n; return f; }**

**3.**

**int f(int a, int b);**

**void main()**

**{ int i=2,p;**

** p=f(i,i+1);**

** printf("%d",p);**

**}**

**int f(int a, int b)**

**{ int c;**

**if(a>b) c = 1;**

**else if(a = = b) c = 0;**

**else c = -1;**

**return(c);**

**}**

**void 6.**

**void fun(int \*a,int \*b)**

**{int \*k;**

**k=a; a=b; b=k;**

**}**

**void main()**

**{int a=10,b=20,\*x=&a,\*y=&b;**

**fun(x,y);**

**printf("%d,%d",a,b);**

**}**

**6. 5.**

**void fun(int \*a,int \*b)**

**{int k; k=\*a; \*a=\*b; \*b=k;}**

**void main()**

**{int a=10,b=20,\*x=&a,\*y=&b;**

**fun(x,y);**

**printf("%d,%d",a,b); }**

**7.**

**void fun(int x,int y)  
{x=x+y;y=x-y;x=x-y; }**

**void main()  
{int x=2,y=3;  
fun(x,y);  
printf("%d,%d\n",x,y);   
}**

**8.**

**void fun(int x,int y,int z)**

**{z=x\*x+y\*y+10;}**

**void main()**

**{int a=100;**

**fun(2,3,a);**

**printf("%d",a);**

**}**

**10.**

**int a=3,b=5;**

**int max(int a, int b)**

**{int c; c=a>b?a:b; return c;}**

**void fmain()**

**{int a=8;**

**printf("%d", max(a,b)); }**

**9.**

**int fun(int n)  
{if(n == 1 || n == 2) return 2;  
return (n-fun(n-1));}**

**vod main()  
{printf(“%d\n”, fun(5));　}**

**12.**

**void func1(int i); void func2(int i);**

**char st[]="hello,friend!";**

**void func1(int i) {**

**printf("%c",st[i]);**

**if(i<3) {i+=2;func2(i);} }**

**void func2(int i) {**

**printf("%c",st[i]);**

**if(i<3) {i+=2; func1(i); } }**

**void main(){**

**int i=0; func1(i); printf("\n"); }**

**11**.设对应的可执行文件为mm.exe

运行时输入：**mm 1 5 9**

**void main(int argc, char \*argv[])**

**{int n=0,i;**

**for(i=1;i<argc;i++)**

**n=n\*10+\*argv[i]-'0';**

**printf("%d\n", n); }**

**13.**

**int func (int a,int b) {**

**static int m=1,i=2;**

**i+=m+1; m=i+a+b;**

**return (m); }**

**void main ( ) {**

**int k=4,m=1,p1,p2;**

**p1=func(k,m) ; p2=func(k,m) ;**

**printf("%d,%d\n",p1,p2) ;**

**}**

**15.**

**void fun(int s[ ]) {  
int i;**

**for(i = 0; i < 10; i ++)**

**printf(“%d”, s[i]);**

**printf(“\n\n”);**

**}**

**void main(){**

**inta[3][4] = {{9,9,2,8},{3,4,5,4},{6,7,8,2}};**

**fun(a[0]);**

**}**

**14.**

**数组a的首地址为1000（十进制数），sizeof（int）的值为4。**

**void main( ) {   
int i,j; int a[2][3]={2,6,10,14,18,20};**

**int \*ptr[2]={a[0],a[1] };**

**for(i=0;i<2;i++)**

**for(j=0;j<3;j++)**

**a[i][j]=\*(ptr[i]+j)/2+a[i][j];**

**printf(“%d\n”,ptr[1]+1);**

**for(i=0;i<2;i++)**

**printf(“%d\n”,\*(ptr[i]+2));**

**}**

**16.**

**void fun(int s[ ]) { int i;**

**for(i = 0; i < 8; i ++)**

**printf(“%d”, s[i]);**

**printf(“\n\n”);**

**}**

**void main(){**

**inta[3][4] = {{9,9,2,8},{3,4,5,4},{6,7,8,2}};**

**fun(a[1]); }**

1. 程序填空

**3. 使用递归函数将一个整数各个位置上的数字顺序输出。例如，输入整数54321，则应输出5 4 3 2 1。**

#include"stdio.h"

void sequence (int n){  
 if(n>10)

(7) ;

printf(“%d ”, (8) );  
}

void main(){

int n; scanf(“%d”,&n); sequence(n);  
}

**4. 在二维字符数组中查找所有包含“abc”的字符串。**

#include "string.h"

void search(char \*p, int n, int i) {  
int j=0;

if(n<3) {

printf(“s[%d]不包含abc”,i);

return;

}

for(;j<n-2;j++)

if( (9) ){

printf(“s[%d]包含abc”,i);

break;

}

if(j==n-2)

printf(“s[%d]不包含abc”,i);  
}

void main( ){

int i;

char s[3][10]={“abc”,”edabcd”,”ssoabd”};

for(i=0;i<3;i++)

search( (10) ,strlen(s[i]),i);  
}

四、编程题目

**3.求10个学生成绩的平均分,最高分和最低分。**

**float average(float array[], int n, float** \*pmax**, float \*pmin) {**

**int i; float aver,sum=array[0];**

**\*pmax= \*pmin =array[0];**

**for(i=1;i<n;i++) {**

**if(array[i]>\*pmax) \*pmax =array[i];**

**else if(array[i]<\*pmin) \*pmin =array[i];**

**sum=sum+array[i];**

**}**

**aver=sum/n; return (aver);**

**}**

**void main( ) {**

**float ave, score[10]={10, ,0,60,70,80, 20,30,40，90,100};**

**int i ; float max, min;**

**ave=average(score,10,&max, &min);**

**printf("……", max,min, ave);**

**}**

1．**写一个判断素数的函数。在主函数输入一个整数，输出是否素数的信息。**

**#include <math.h>**

**int prime(int);**

**void main()**

**{int m,flag;**

**scanf("%d", &m);**

**flag=prime(m);**

**if(flag==1) printf("%d is a prime ",m);**

**else printf("%d is not a prime ",m);**

**}**

**int prime(int m)**{

**int i,k,result;**

**k=sqrt(m);**

**for(i=2;i<=k;i++)**

**if(m%i==0) break;**

**if(i==k+1) result=1;**

**else result=0;**

**return result;**

**}**

**4.有n个数，使前面各数顺序向后移m个位置，最后m个数变成最前面m个数。  
在main中输入n个数和输出调整后的n个数。(递归算法)**

**void move(int array[],int n,int m);**

**void main()**

**{int number[20],n,m,i;**

**scanf("%d", &n); scanf("%d",&m);**

**for(i=0;i<n;i++)**

**scanf("%d",&number[i]);**

**move(number,n,m);**

**for(i=0;i<n;i++)**

**printf("%d ",number[i]);**

**}**

**void move(int array[ ],int n,int m) {**

**int \*p, array\_end;**

**array\_end=\*(array+n-1);**

**for(p=array+n-1;p>array;p--)**

**\*p=\*(p-1);**

**\*array=array\_end;;**

**m--;**

**if(m>0) move(array,n,m);**

**}**

**2.写一个函数，将两个字符串连接。**

**void str\_connect(char[], char[]);**

**void main( ) {**

**char str1[20],str2[20];**

**gets(str1); gets(str2);**

**str\_connect(str1,str2);**

**puts(str1);**

**}**

**void str\_connect(char s1[], char s2[])**

**{int i;int j;**

**i=0;**

**while(s1[i]!='\0')**

**i++;**

**j=0;**

**while(s2[j]!='\0')**

**s1[i++]=s2[j++];**

**s1[i]='\0';**

**}**

考虑在函数str\_connect中通过指针形式如何实现？

**教材例题[5.3] [ 5.4] [5.6] [5.6]换成选择排序、[5.8]、[5.10] [5.13] [5.14]**

**教材习题 [5.2]**

**一、选择题**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **D** | **C** | **A** | **D** | **B** | **C** | **C** | **A** | **D** | **C** |

**二、写出程序运行结果**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **AA** | **28** | **-1** | **-1 -4** | **20,10** | **10,20** | **2,3** | **100** |
| **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **2** | **8** | **159** | **hlo** | **9,19** | **1016**  **15**  **30** | **9 9 2 8 3 4 5 4 6 7** | **3 4 5 4 6 7 8 2** |

C语言概念题解答选编

**单项选择题**

1. 下列字符列中，不是C语言提供的合法关键字是

① switch ② begin ③ case ④ default

【解】因C语言的关键字表中没有begin，它不是C语言的关键字。所以解答是②。

1. 下列字符列中，不是用来表达转义字符是

① \\ ② \’ ③ 074 ④ \0

【解】转义字符有三种书写形式：反斜杠字符后接上某个规定的字符；反斜杠字符后接上1至3个八进制数字符；反斜杠字符和字符x之后接上1至2个十六进制数字符。后两种分别以八进制数和十六进制数直接给出字符的ASCII代码值。而074是八进制整数，不是转义字符。所以解答是③。

1. 字符串 ″xyz″在内存占用的字节数是

① 3 ② 4 ③ 6 ④ 8

【解】字符串中的每个字符占1个字节，但C程序在存储字符串时，要在最后一个有效字符后面接上1个字符串结束的标记符’\0’。这样，存储字符串常量″xyz″需要4个字节。所以解答是②。

1. 设变量m，n，a，b，c，d均为0，执行(m = a==b)||(n=c==d)后，m，n的值是

① 0，0 ② 0，1 ③ 1，0 ④ 1，1

【解】计算(m = a==b)||(n=c==d)的过程是先计算逻辑或的左运算分量(m = a==b)的值，由于赋值运算符的优先级比关系运算符==的优先级低，又先计算a==b。因a，b均为0后，所以比较结果值为1。将1赋给变量m，使变量m的值变为1。同时这个赋值运算的结果也是1，这样逻辑运算的左运算分量的值为1。由于逻辑或运算的左运算分量值为1，按照C语言对逻辑或运算优化计算的规定，不再计算逻辑或的右运算分量，而直接得到逻辑或运算的结果为1。由于逻辑或的右运算分量不曾计算过，所以变量n的值也不为变化。这样，上述表达式计算后，变量m 的值为1，变量n的值为0。所以解答是③。

1. 设整型变量m，n，a，b，c，d均为1，执行“(m=a>b)&&(n=a>b)”后，m，n的值是

① 0，0 ② 0，1 ③ 1，0 ④ 1，1

【解】表达式(m=a>b)&&(n=a>b)是一个逻辑表达式，它的计算过程是先计算逻辑与的左分量(m=a>b)，其中又是先计算a>b。因a>b不成立，结果为0，将0赋给变量m，最后逻辑与的左分量结果为0。由于逻辑运算采用特别的优化计算规则，当逻辑与的左分量结果为0时，不再计算逻辑与的右分量，而直接以0为逻辑与的结果。所以，上述表达式计算使m的值变为0，而n的值没有变，依旧为1。所以解答是②。

1. 设有语句“int a=3;”，则执行了语句“a + = a - = a\*a；”后，变量a的值是

① 3 ② 0 ③ 9 ④ -12

【解】由于赋值运算符的结合性自右至左，语句“a + = a - = a\*a；”的执行是先计算a\*a，得到值9，再计算a - = a\*a，使a 的值为-6，接着计算a += a，使a 的值为-12。所以解答是④。

1. 在以下一组运算符中，优先级最高的运算符是

① <= ② = ③ % ④ &&

【解】常规运算符的优先级自高到低的排列顺序是算术运算符、移位运算符、关系运算符、按位运算符、逻辑运算符、条件运算符、赋值运算符、逗号运算符。所以问题所给的四个运算符按优先级自高到低的排列顺序是%，<=，&&，=。所以解答是③。

1. 若已定义x和y为double类型，则表达式：x=1, y = x+3/2的结果是

① 1 ② 2 ③ 2.0 ④ 2.5

【解】由于变量x和y都是double类型，表达式x=1, y = x+3/2是先计算x=1，由于1是整型的，x是double类型的，该计算先将整数1转换成double类型的1.0L，并将1.0L赋给变量x。计算y = x+3/2是先求3/2的值，因是两整数相除，结果是整型值1，然后在计算x+1时，也先将1转换成1.0L，然后求得x+1的值为2.0L，最后将该值赋给变量y，并得到整个表达式的值为2.0L，简写成2.0。所以解答是③。

1. 设a=1，b=2，c=3，d=4，则表达式a < b ? a : c < d ? a : d的结果为

① 4 ② 3 ③ 2 ④ 1

【解】条件运算符的优先级比关系运算符的优先级低，并且它的结合性是自右向左的，所以表达式a < b ? a : c < d ? a : d可用圆括号等价地写成(a < b) ? a : ((c < d) ? a : d)。因a<b成立，计算结果为a的值1。所以解答是④。

1. 设a为整型变量，下列C表达式中，不能表达数学关系式：10<a<15的是

① 10<a<15 ② a == 11 || a == 12 || a == 13 || a == 14

③ a>10&&a<15 ④ !(a<=10)&&!(a>=15)

【解】数学关系式10<a<15表示a只能是11，12，13，14四个值之1。用C表达式表示这个条件，可写成a >10&& a < 15，或!(a<=10)&&!(a>=15)，或a == 11 || a == 12 || a == 13 || a == 14等。若写成10<a<15，该算式的计算顺序是先求出10<a的结果0或1，然后以这个结果与15比较，是否小于15，显然这与数学式子的意义是不一致的。所以解答是①。

1. 若有以下定义：char a; int b; float c; double d;则表达式a\*b+d-c值的类型为

① float ② int ③ char ④ double

【解】基本类型的数据可以混合运算，但运算时，对要求运算分量类型相同的运算来说，如运算分量的类型不是全相同，则会有自动类型转换发生。类型自动转换规则规定类型低（简单）的向类型高（复杂）的转换。各类型由高到低的排列顺序是：

long double 、double、float、unsigned long、long、unsigned int、int。

这样，若有定义：char a; int b; float c; double d;则表达式a\*b+d-c的值的类型是double的。首先计算a\*b，得到int类型的值，接着计算a\*b+d，得到double类型的值，最后计算a\*b+d-c，得到double类型的值。所以解答是④。

1. 表达式“9！= 7”的值是
2. true ② 非0值 ③ 0 ④ 1

【解】关系运算的结果只有0和1，若关系成立，则结果为1；若关系不成立，则结果为0。因关系式9！= 7成立，所以表达式的结果为1。虽然1也是非0值，在供选择的答案中有1，所以最确切的解答是④。

1. 用十进制数表示表达式“12 | 012”的运算结果是

① 1 ② 0 ③ 14 ④ 16

【解】按位运算将运算分量视作二进位的位串信息，对两运算分量的对应位进行运算，产生二进位的位串信息结果。整数12最低8位二进位是00001100，整数012是八进制数，它的最低8位二进位是00001010。这两个二进位位串信息按位或运算，得到结果是00001110，对应十进制数14，八进制数是016。所以解答是③。

1. 设字符型变量a=3，b=6，计算表达式c = (a^b)<<2后c 的二进制值是

① 00011100 ② 00000111 ③ 00000001 ④ 00010100

【解】a的值为3，写成8位二进制数形式是00000011，b的值为6，写成8位二进制数形式是00000110。表达式c = (a^b)<<2，先计算a与b的按位加，得到结果00000101，将这结果向左移2位，得到二进位的位串信息是00010100。所以解答是④。

．设ch是char型变量，其值为’A’，则下面表达式的值是

ch = (ch>=’A’&&ch<=’Z’)?(ch-‘A’+’a’):ch

① A ② a ③ Z ④ z

【解】由于字符型变量ch的值为’A’，计算表达式ch = (ch>=’A’&&ch<=’Z’)?(ch+32):ch，先计算其中条件表达式，由于条件(ch>=’A’&&ch<=’Z’)成立，该条件表达式以ch+32=97为结果，将该值赋给变量ch，以字符表达这个值为’a’。 所以解答是②。

28．设a=3，b=4，执行“printf(“%d, %d”, (a, b), (b, a));”的输出是

1. 3，4 ② 4，3 ③ 3，3 ④ 4，4

【解】在调用格式输出函数的语句中，其中每个格式符对应一个输出项，格式符d要求输出项内容以十进制整数形式输出。语句中的第一个格式符对应输出项(a, b)是一个逗号表达式，该表达式的值是b 的值，所以先输出4。接着输出字符逗号。同样输出项(b,a)的值是a的值，输出3。所以语句执行将输出4, 3。正解的选择是②。

32．设有int i=010, j=10；则执行“printf(“%d, %d\n”, ++i, j--)；”的输出是

① 11，10 ② 9，10 ③ 010，9 ④ 10，9

【解】变量i和j的初值分别为八进制数010和十进制数10，格式输出函数调用printf(“%d, %d\n”, ++i, j--)中，++i的值是变量i增1后的值，原来值是八进制数010，等于十进制数8，输出9。j--的值是变量j减1之前的值，输出10。格式字符串中的逗号是普通字符照原样输出。所以问题给出的格式输出函数调用将输出9,10。正确的解答是②

1. 假定所有变量均已正确定义，下列程序段运行后x的值是

a = b = c = 0; x = 35;

if(!a)x--;else if(b);if(c)x = 3;else x = 4;

① 34 ② 4 ③ 35 ④ 3

【解】以变量a，b，c的值均为0，变量x的值为35，语句：

if(!a)x--;else if(b);if(c)x = 3;else x = 4;

由两个if语句组成。首先执行前一个if语句“if(!a)x--;else if(b)；”，因变量a的值为0，条件!a成立，执行x--使x的值变为34。接着执行后继的if语句“if(c)x = 3;else x = 4;”，因变量c的值为0，条件不成立而执行x = 4，最终使变量x的值为4。正确解答是②。注意前一个if语句的else部分的成分语句只有“if(b)；”，这是一个单分支if语句，且其成分语句为空语句。

1. 下面的程序段所表示的数学函数关系是

y = -1;

if(x!=0) if(x>0) y = 1; else y = 0;

-1 (x<0) 1 (x < 0) 0 (x<0) -1 (x<0)

① y = 0 (x=0) ② y = -1 (x = 0) ③ y = -1 (x = 0) ④ y = 1 (x = 0)

1 (x>0) 0 (x > 0) 1 (x > 0) 0 (x > 0)

【解】程序段首先置变量y的值为-1，接着按变量x值的不同情况重置变量y的值。重置的首要条件是x!=0，若变量x的值为0，则不再重置变量y的值，所以在x值为0情况下，y的值是-1。在变量x的值不等于0的条下，若x的值大于0，重置变量y的值为1；若变量x的值小于0，重置变量y的值为0。所以程序段实现当变量x的值为0时，变量y的值为-1；当变量x的值大于0时，置变量y的值为1；当变量x的值小于0时，置变量y的值为0。正确解答为③。

38．执行下列程序段

x = 9;

while (x > 7)

{ printf(“\*”); x--; }

后输出的结果是

① \*\*\*\* ② \*\*\* ③ \*\* ④ \*

【解】上述代码以x的初值为9，在x>7成立的情况下循环，每次循环输出一个\*字符，并让x的值减1。共执行2次循环，也就共输出了2个\*字符。所以解答为③。

1. 下列语句中，错误的是
2. while (x=y)5; ② do x++ while(x==10);

③ while(0); ④ do 2; while (a==b);

【解】while语句的一般形式是：

while(表达式)语句

这里的表达式可以是任何合理的表达式，语句可以是任何语句，包括空语句，或表达式语句。可能书写的while语句没有任何意义，如供选择的①while(x=y)5;和③while(0);但这两个语句的句法没有任何错误。do-while语句的一般形式是：

do 语句while (表达式)；

其中的成分语句也可以是任何语句，但供选择答案②dox++while(x==10);中的代码x++是一个表达式，不是语句，所以是错误的。正确解答是②。

1. 循环语句“for（x=0, y=0; (y!=123) || (x < 4); x++）;”的循环执行
2. 无限次 ② 不确定次 ③ 4次 ④ 3次

【解】for循环语句的初始化部分置变量x和y的初值为0，循环条件是(y!=123) || (x < 4)，每次循环后变量x的值增1。由于循环过程中变量y的值未被修改过，循环条件又是一个逻辑或，其左分量(y!=123)永远成立，也就是循环条件永远成立。所以该循环语句将循环执行无限次。正确解答是①。

41．若i、j已定义为int类型，则以下程序段中的内循环体的执行次数是○。

for(i = 5; i; i--)

for(j = 0; j < 4; j++) { …… }

1. 20 ② 24 ③ 25 ④ 30

【解】问题所给的程序段的外循环是一个for循环语句，它的初始化部分置变量i的初值为5，循环条件简写成i，即i!=0，每次循环后变量i的值减1。所以外循环共控制5次循环。内循环也是一个for循环语句，它的初始化部分置变量j的初值为0，循环条件是j<4，每次循环后变量j的值增1。所以内循环共控制4次循环。这样，如内循环的体内未曾对变量i和j有修改，在内、外循环一起控制下，内循环的体共被重复执行20次。正确解答是①。

42．假定a和b为int型变量，则执行以下语句后b的值为

a = 1; b = 10;

do { b -= a; a++;

} while (b-- < 0);

① 9 ② -2 ③ -1 ④ 8

【解】在这个程序段中，循环开始前变量a 的值为1，b的值为10，每次循环从变量b减去a，并让a增1，并在循环条件判定时，又让b减去1。第一次循环后，变量b的值变成9，变量a 的值变为2，循环判断时，因b的值大于0，循环条件不成立，结束循环。但在循环判断时，让b减去了1，所以循环结束时，变量b的值为8。正确的解答是④。

1. 设x和y为int型变量，则执行下面的循环后，y的值为

for(y=1, x=1; y <=50; y++) {

if(x >= 10) break;

if(x%2 == 1) { x += 5; continue;}

x -= 3;

}

① 2 ② 4 ③ 6 ④ 8

【解】for循环语句的初始化部分置变量x和y的初值为1，循环条件是(y<=50)，每次循环后变量y的值增1，控制循环最多执行50次。循环体有三个语句：首先在发现变量x的值大于等于10时，结束循环；接着是当变量x除2的余数为1（即变量x是奇数）时，让变量x值增5，让x变成偶数，并直接进入下一轮循环；如变量x是偶数，则从变量x减去3，让变量x变成奇数。由上述分析知，每两次循环使变量x的值增加2。第一次循环后，变量x的值变成6。第二次循环后，变量x的值变成3。第三次循环后，变量x的值变成8。第四次循环后，变量x的值变成5。第五次循环后，变量x的值变成10。第六次循环时，因变量x的值大于等于10，直接跳出循环，这次循环是非正常结束，对变量y的修正只执行了5次。所以循环结束后，变量y的值增至6。正确的解答是③。

1. 在C语言中，下列说法中正确的是
2. 编程时尽量不要使用“do 语句 while（条件）”的循环

② “do 语句 while（条件）”的循环中必须使用“break”语句退出循环

③ “do 语句 while（条件）”的循环中，当条件非0时将结束循环

④ “do 语句 while（条件）”的循环中，当条件为0时将结束循环

【解】do-while语句的一般形式是：

do 语句

while (表达式)；

其语义是重复执行其成分语句，直至表示条件的表达式值为0时结束。do-while语句是正常使用的一种循环结构之一。do-while语句的循环结束条件由while 后的表达式值为0所控制，并不一定要有break语句跳出循环来结束循环。do-while语句在条件值非0时，将继续循环，而不是结束循环。条件值为0时，才结束循环。所以正确的选择是④。

1. 若有下列说明，则数值为4的表达式是

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

char c = ‘a’, d, g;

1. a[g-c] ② a[4] ③ a[‘d’-‘c’] ④ a[‘d’-c]

【解】数组元素的下标自0开始顺序编号，值为4的元素是a[3]。所以只有下标表达式的值为3才是正确的。下标表达式g-c中的的变量g和c的值是还未被设定的，其值不确定。a[4]的值为5，下标为4是不对的。’d’-‘c’的值为1，a[1]的值是2，所以也是不对的。变量c的值为’a’，’d’-c=3，所以正确解答是④。

1. 设有定义：“char s[12]={“string”};”，则printf(“%d\n”, strlen(s));的输出是
2. 6 ② 7 ③ 11 ④ 12

【解】在C语言中，字符串是指在有效字符之后有字符串结束标记符的字符列，并约定字符串的长度是指字符串中有效字符个数，不包括字符串的结束标记符。存于字符数组s中的字符串是”string”，该字符串的长度为6，所以输出该字符串的长度应是6。正确的解答是①。

1. 下列C代码中，正确的是
2. char a[3][ ] = {‘abc’, ‘1’}; ② char a[ ][3] = {‘abc’, ‘1’};

③ char a[3][ ] = {‘a’, “1”}; ④ char a[ ][3] = {“a”, “1”};

【解】如定义数组时有初始化，其最高维的长度可以省缺，由系统根据初始化时的初值个数确定，而其它维的长度是不可以缺省的。对两维字符数组的初始化，可以按行用字符串对其初始化，也可以按两维数组元素的存储顺序逐一用字符对其元素初始化。在供选择解答①和③中，有不是最高维的长度被缺省，所以是错误的。在供选择解答①和②中，还将字符串常量写作’abc’，这也是错误的。只有④，用字符串按行给两维字符数组初始化，这才是正确的。正确的解答是④。

50．字符串”m\x43\\\np\102q”的长度是

① 7 ② 9 ③ 15 ④ 16

【解】字符串的长度是指字符串中有效字符的个数。在C程序中，组成字符串的字符可以是一般的普通字符，也可以是转义字符。其中转义字符又可以用多种不同形式来表达：反斜杠字符之后接上预先规定的字符，用来表示特殊字符或特定意义的控制字符。如单引号字符、双引号字符、反斜杠字符等特殊字符，换行、回车、退格等控制字符。反斜杠字符之后接上1至3位八进制数字符，这是直接用八进制数字表示字符的ASCII代码的方式。反斜杠字符之后接上字符x，并后接1至2个十六进制数字符，这是直接用十六进制数字表示字符的ASCII代码的方式。后二种能表示C语言允许的任何字符。在本小题给出的字符串中，字符’\x43’是用十六进制数表示的一个字符，字符’\\’表示一个反斜杠字符，字符’\n’表示一个换行字符，字符’\102’是用八进制数表示的一个字符。这样，所给字符串只有7个有效字符。所以，该字符串的长度是7，解答为①。

55．设char str1[10] = “ABCDE”, str2[10] = “xyz”;

则执行语句 printf(“%d”, strlen(strcpy(str1, str2))); 后的输出结果是

① 9 ② 8 ③ 5 ④ 3

【解】上述代码是输出表达式strlen(strcpy(str1, str2))的值，该表达式的求值 顺序是先执行函数调用strcpy(str1, str2)，该函数调用将str2 所指字符串内容复制到str1所指的字符数组中，并返回存放于str1中的字符串的首字符的指针。经函数调用strcpy(str1, str2)后，str1中的内容变成字符串“xyz”。接着求更新成字符串“xyz”的str1中的字符串的长度，并输出。这样，代码将输出整数3。所以解答为④。

56．

58．在C语言中，下列说法中，错误的叙述是

1. 函数定义可以分为两个部分：函数说明部分和函数体
2. 主函数可以调用任何非主函数
3. 任何非主函数可以调用其它任何非主函数
4. 程序可以从任何函数开始执行

【解】每个C函数的定义分两部分，函数说明部分和函数体，所以叙述①是正确的叙述。在C语言中，函数可以递归调用，主函数可以调用程序中的任何函数，当然可以调用任何非主函数的其它函数，所以叙述②是一个正确的叙述。同样理由，叙述③也是正确的。C语言规定，C程序只有一个主函数，并总是从主函数开始执行，不能从非主函数开始执行。所以，说程序可以从任何函数开始执行是错误的。所以解答是④。

59．

60．执行下列程序

int fun(int x1, int x2)

{ int x;

x1 > x2 ? (x = 3) : (x = 4);

return x + x1;

}

main()

{ printf(“%d\n”, fun(7, 8));

}

后输出结果是

① 9 ② 10 ③ 11 ④ 12

【解】上述程序中，主函数以实际参数7和8调用函数fun，并输出函数的返回值。函数fun被调用时，形式参数x1的初值是7，x2的初值是8。函数执行表达式语句

x1 > x2 ? (x = 3) : (x = 4)

这是一个条件表达式，因条件x1 > x2不成立，条件表达式求(x = 4)，该表达式的求值使函数的局部变量x的值为4。最后，函数返回x + x1的值11。所以解答为③。

1. 若有说明语句“int a[10], \*p = a;”，对数组元素的正确引用是

① a[p] ② p[a] ③ \*(p+2) ④ p+2

【解】在C语言中，约定数组名单独出现在表达式中时，它表示数组首元素的指针。有int a[10]，则a可以作为&a[0]使用。另有整型指针变量p，代码p=a实现p指向数组a的首元素。则表达式\*(p+2)是引用数组元素a[2]。表达式a[p]和p[a]都是不正确的，下标必须是整型表达式，不可以是指针表达式。表达式p+2是指针表达式，它的值是&p[2]。所以只有表达式\*(p+2)引用数组a的元素a[2]。所以解答是③。

1. 下面各语句中，能正确进行赋字符串操作的语句是

① char s[5] = {“ABCDE”}; ② char s[5] = {‘A’,’B’,’C’,’D’,’E’};

③ char \*s; s = “ABCDE”; ④ char \*s; scanf(“%s”, s);

【解】字符串最终存储于字符数组中，存储字符串的字符数组可以是程序主动引入的（定义或动态分配），也可以是由系统分配的。其中字符数组用字符串常量初始化就是字符串存储于由程序引入的字符数组的例子。给字符指针赋字符串则是系统自动分配字符串存储空间的例子。给字符指针赋字符串常量并不是将一个长长的字符串常量存于字符指针变量中，而是将字符串常量存储于常量区，并将存储这个字符串常量的首字节地址赋给指针变量，让指针变量指向字符串常量的首字符。对于以字符串常量作为字符数组初值的情况，要求字符数组足够的大，能存得下字符串常量。这里有一个特别的规定，若数组的大小少于存储字符串有效字符的字节个数，系统将报告错误；当字符数组的大小只能存储字符串的有效字符，而不能存储字符串结束标记符时，则存储于字符数组中的内容是字符序列，因没有存储字符串结束标记符，存储的内容就不是字符串。如代码char a[5] = “ABCDE”。另外，给字符数组元素逐一赋字符初值，并在字符初值中没有字符串结束标记符，则存于字符数组中的内容也不是字符串。如代码char s[5] = {‘A’,’B’, ‘C’, ‘D’, ‘E’}。特别要注意当字符指针还未指向某个字符数组的元素时，不可以通过字符指针输入字符串。如代码char \*s; scanf(“%s”, s)。若写成char \*str; scanf(“%s”, &str)更是错误的了。由于C语言规定数组不能相互赋值，所以只能将字符串常量赋给某字符指针。如代码char \*s; s = “ABCDE”是正确的。实际上，字符串”ABCDE”被存储于常量区中，向指针变量赋的是字符指针，让s指向其中的字符’A’。 所以解答是③。

1. 若有以下定义，则数值为4的表达式○。

int w[3][4] = {{0, 1}, {2, 4}, {5, 8}}, (\*p)[4] = w;

① \*w[1]+1 ② p++, \*(p+1) ③ w[2][2] ④ p[1][1]

【解】两维数组定义有多种赋初值的办法，问题给出的代码是按行给数组的部分元素赋初值。它们分别是w[0][0] = 0、w[0][1]=1、w[1][0]=2、w[1][1] = 4、w[2][0]=5、w[2][1]=8。根据约定，未指定初值的元素自动置全0值。指针变量p是一个指向具有四个int型元素的一维数组的指针，定义时的初值使它指向数组w的第一行。①的代码，\*w[1]+1中的w[1]是指向w[1][0]的指针，\*w[1]就是w[1][0]，其值是2，\*w[1]+1的值是3。②的代码是逗号表达式，“p++，\*(p+1)”先使p指向数组w的第二行，\*(p+1)中的p+1是指向数组w的第三行，\*(p+1)是指针值，指向数组w的第三行的第一个元素，即&w[2][0]。③的代码w[2][2]引用数组w第三行的第三列元素，其值是0。④的代码p[1][1]引用数组w第二行的第二列元素w[1][1]，其值是4。所以解答是④。

1. 若有下面的程序片段，则对数组元素的错误引用是○。

int a[12] = {0}, \*p[3], \*\*pp, i;

for(i = 0; i < 3; i++) p[i] = &a[i+4];

pp = p;

① pp[0][1] ② a[10] ③ p[3][1] ④ \*(\*(p+2)+2)

【解】上述代码定义变量a是有12个整型元素的数组，它的初值全是0。p是有3个元素的指针数组，每个指针能指向整型变量。pp是指针的指针，它能指向一个指向整型变量的指针，i是一个整型变量。执行代码for(i = 0; i < 3; i++) p[i] = &a[i\*4]使指针数组p的元素p[0]指向a[4]，p[1]指向a[5]，p[2]指向a[6]。代码pp = p使指针变量pp指向p[0]。①，代码pp[0][1]用指针表达可等价地写成\*(\*pp+1)。其中\*pp就是p[0]，p[0]的值是&a[4]，\*pp+1的值是&a[4]+1 = &a[5]，\*(\*pp+1)就是a[5]。②，代码a[10]当然是引用数组a的元素。③，代码p[3][1]因数组p只有三个元素，没有p[3]，所以是错误的。④，代码\*(\*(p+2)+2)中的(p+2)是&p[2]，\*(p+2)是p[2]，其值是&a[6]，\*(p+2)+2的值是&a[8]，\*(\*(p+2)+2)引用的是a[8]。所以解答是③。

1. 若有如下定义和语句，则输出结果是○。

int \*\*pp, \*p, a = 10, b = 20;

pp = &p; p = &a; p = &b; printf(“%d, %d\n”, \*p, \*\*pp);

① 10, 20 ② 10, 10 ③ 20, 10 ④ 20, 20

【解】上述代码定义变量pp是指针的指针，它能指向一个指向整型变量的指针。定义变量p是指针，它能指向一个整型变量。a是一个整型变量，它有初值10。b也是一个整型变量，它有初值20。代码pp=&p使pp指向p，p = &a使p指向a，p = &b又使p指向b，不再指向a。所以\*p是引用b，其值为20。\*\*pp是通过pp间接引用p，再通过p间接引用b，所以也是20。所以解答是④。

1. 若有以下定义和语句，则对w数组元素的非法引用是○。

int w[2][3], (\*pw)[3]; pw = w;

① \*(w[0]+2) ② \*pw[2] ③ pw[0][0] ④ \*(pw[1]+2)

【解】上述代码定义2行3列的两维数组w，定义能指向有三个整型元素一维数组的指针pw，并让pw指向数组w的第一行。①，代码\*(w[0]+2)中的w[0]是&w[0][0]，w[0]+2是&w[0][2]，所以，\*(w[0]+2)就是w[0][2]。②，代码\*pw[2]中的pw[2]是&w[2][0]，该数组w只有2行，没有w[2][0]，所以代码\*pw[2]是对数组w元素的非法引用。③，代码pw[0][0]就是w[0][0]。④，代码\*(pw[1]+2) 中的pw[1]就是\*(pw+1)，即&w[1][0]，pw[1]+2就是&w[1][2]，所以，\*(pw[1]+2)就是w[1][2]。所以解答是②。

72．函数调用语句“f((e1, e2), (e3, e4, e5))；”中参数的个数是○。

① 1 ② 2 ③ 4 ④ 5

【解】上述函数调用中，(e1, e2)和(e3, e4, e5)是两个带括号的表达式，所以函数调用只提供两个实参，其中第一个实参先计算e1的值，然后计算e2，并以e2的值为实参。第二个实参顺序计算e3，e4，e5，并以e5的值为实参。所以解答是②。

73．C语言中，函数的隐含存储类型是○。

① auto ② static ③ extern ④ 无存储类别

【解】由于C函数内不能再定义函数，C函数的存储类型只能是静态的或外部的之一。若定义函数时不指定存储类型是静态的，则它的存储类型就是外部的（extern），即函数允许被程序的其它函数调用。所以解答是③。

74．以下对C语言函数的有关描述中，正确的是○。

1. 在C语言中，调用函数时，只能把实参的值传递给形参，形参的值不能传递给实参
2. C函数既可以嵌套定义，又可以递归调用
3. 函数必须有返回值，否则不能使用函数

④ C程序中，有调用关系的所有函数必须放在同一个源程序文件中

【解】C语言规定，调用函数时，只能把实参的值传递给函数的形参。函数调用时，形参从实参表达式得到初值，形参也是函数的一种局部变量，其值可以改变，但形参的值不能传回给对应的实参。当函数设置非指针类型的形参时，实参可以是同类型的一般表达式；当函数设置指针类型的形参时，对应的实参也必须是同类型的指针表达式。所以叙述①是正确的。C函数虽可以递归调用，但同时规定，在函数内不能再定义函数，所以叙述②是错误的。通常C函数会有返回值，但也可以没有返回值，有许多情况，函数的执行是完成某种预定的工作，并没有返回值，所以叙述③是不正确的。在C程序中，如函数未指定是静态的，则就是外部的，能让别的文件中的函数调用，但函数要调用别的文件中的函数，在调用代码之前，需对它作说明，所以叙述④也是不正确的。正确的解答只有①。

75．设函数的功能是交换x和y的值，且通过正确调用返回交换结果。不能正确执行此功能的函数是○。

1. funa(int \*x, int \*y)

{ int \*p;

\*p = \*x; \*x = \*y; \*y = \*p;

}

② funb(int x, int y)

{ int t;

t = x; x = y; y = t;

}

③ func(int \*x, int \*y)

{ int p;

p = \*x; \*x = \*y; \*y = p;

}

④ fund(int \*x, int \*y)

{ int \*p;

\*x = \*x + \*y; \*y = \*x - \*y; \*x = \*x - \*y;

}

【解】设计实现交换两个变量值的函数。希望函数调用能改变调用环境中的变量，方法之一是让函数设置指针类型的形参，实参传送给形参的是希望交换值的两个变量的指针。函数的两个形参得到这两个变量的指针后，函数就能通过形参间接引用要交换值的变量，或引用它们的值，或改变它们的值。从而实现从函数传回信息存于调用环境的变量中。以上所述是设计交换变量值函数的一般原则，具体编写函数时还需注意对上述原则的正确应用。

对于函数funa，该函数的定义满足设置指针形参，并通过指针形参间接引用环境变量的要求。但是在函数体中定义了指针变量p，并在p未明确设定它所指向的变量下，代码\*p = \*x企图将值存入它所指的变量中，这是初学者经常出现的一种严重的错误。该函数因有错，不能实现交换变量值的功能。

对于函数funb，该函数不定义指针形参，函数体也没有可用的指针间接引用调用环境中的变量，所以该函数也不能实现交换变量值的功能。

对于函数func，该函数正确定义了指针形参，函数体也正确通过指针形参间接引用环境中的变量，并正确定义了自已的局部变量。该函数能实现交换变量值的功能。

对于函数fund，该函数也正确定义指针形参和函数体通过形参间接引用环境变量。特别要指出的是，该函数是利用\*x和\*y的旧值与新值之间的关系实现问题要求的。其中第一个赋值使\*x存有原来未交换之前的\*x与\*y之和；第二个赋值从这个和中减去原来的\*y，并将差赋给\*y，使\*y的值是交换之前的\*x的值；第三个赋值再次从和中减去新的\*y，即减去交换之前的\*x，这个差就是交换之前的\*y，并将差赋给\*x。经以上三个赋值，实现两变量之间的值交换。所以不能正确执行交换变量值的函数是funa和funb，即正确选择是①和②。

1. 要求函数的功能是在一维数组a中查找x值。若找到，则返回所在的下标值；否则，返回0。设数据放在数组元素的a[1]到a[n]中。在以下给出的函数中，不能正确执行此功能的函数是○。

① funa(int \*a, int n, int x)

{ \*a = x;

while(a[n] != x) n--;

return n;

}

② funb(int \*a, int n, int x)

{ int k;

for(k = 1; k <= n; k++)

if(a[k] == x) return k;

return 0;

}

③ func(int a[], int n, int x)

{ int \*k;

a[0] = x; k = a+n;

while(\*k != x) k--;

return k – n;

}

④ fund(int a[], int n, int x)

{ int k = 0;

do k++;

while((k < n+1)&&(a[k]!=x));

if((k < n+1)&&a[k] == x)) return k;

else return 0;

}

【解】在数组中找指定值是经常遇到的计算要求，有多种编程方法。在这里，数据预放在数组下标1至n的元素中，下标为0的元素没有放数据，程序可以利用这个位置简化查找函数。函数funa先将要查找的值放入a[0]，从数据表的最后一个元素开始逆序向前查找。这样做的好处是循环条件不必担心因数组中原先没有值为x的元素而一直向前，访问不是数表的元素，需插入条件n>0。在a[0]处放入x后，这个条件就不必要了，循环至少在访问了a[0]后终止，并返回0值。所以该函数能完成指定的功能。函数funb采用常规的办法编写，循环在a[1]与a[n]之间顺序寻找，一旦找到立即返回找到处的下标，直至查找循环结束，查不到指定的值时，返回0值。函数func采用与函数funa相同的方法，不过是另引入一个指针变量。但是该函数return语句后的表达式有严重的错误，应返回k-a，两指针的差是一个整数，在这里是找到元素的下标。表达式k-n是指针k向前移n个位置的指针值。函数fund预置k为0，循环让k增1，并在k在界内和a[k]不等于x的情况下继续循环。循环结束有两种情况，或k已不在界内，或k在界内，并且a[k]等于x。若是后者，函数返回k，而若前者，函数返回0。该函数也能正确完成查找工作。这样，不能正确完成查找工作的函数是函数func。所以正确的选择是③。

1. 以下程序的输出结果是○。

#include <stdio.h>

sub1(char a, char b){ char c; c = a; a = b; b = c;}

sub2(char \*a, char b){ char c; c = \*a; \*a = b; b = c;}

sub3(char \*a, char \*b){ char c; c = \*a; \*a = \*b; \*b = c;}

main()

{ char a, b;

a = ‘A’; b = ‘B’; sub3(&a, &b); putchar(a); putchar(b);

a = ‘A’; b = ‘B’; sub2(&a, b); putchar(a); putchar(b);

a = ‘A’; b = ‘B’; sub1(a, b); putchar(a); putchar(b);

}

① BABBAB ② ABBBBA ③ BABABA ④ BAABBA

【解】在上述程序中，函数sub1完成两形参值的交换，与实参提供的变量无关，这是一个没有意义的函数。函数sub2将第二个形参的值置入由第一个指针形参所指的变量中，指针形参所指的变量由调用时的实参提供。函数sub3完成将两个形参所指的变量的值交换的工作。程序调用sub3，使变量a和b的值交换输出BA；调用sub2，使变量b的值传送到a，输出BB；调用sub1，变量a和b的值不改变，输出AB。所以程序输出BABBAB。正确解答是①。

1. 设有定义语句“int (\*ptr)[10];”，其中的ptr是
2. 10个指向整型变量的指针
3. 指向10个整型变量的函数指针

③ 一个指向具有10个元素的一维数组的指针

④ 具有10个指针元素的一维数组

【解】代码“int (\*ptr)[10];”的分析过程是，因圆括号，括号内的ptr先与字符\*结合，字符\*修饰标识符ptr，使它是一种指针；接着与后面的一对方括号结合，表示是这样一种指针，是指向一维数组的；再有方括号中的10，说明这种数组有10个元素。至此，ptr是指向含10个元素的一维数组的指针。最后，是最前面的int，表示数组元素是int类型的。所以，ptr是一个指向具有10个int型元素的一维数组的指针。所以解答是③。另外，要是①，10个指向整型变量的指针，就是一个指针数组，上述代码应写成“int \*ptr[10];”，即ptr是一个有10个指向整型变量的数组。 要是②，返回值是指向10个整型变量的函数的指针，上述代码应写成“int (\*(\*ptr)())[10];”，即ptr是指针，指向一种函数，函数的返回值又是一种指针，指向10个元素的数组，数组的元素是整型的。下面的代码是这样的函数指针和函数的例子：

#include <stdio.h>

int a[][10] = {{1,2,3,4,5,6,7,8,9,0},{0,1,2,3,4,5,6,7,8,9}};

int (\*(\*ptr)(int))[10];

int (\*f(int n))[10]

{ return a+n;

}

void main()

{ int (\*p)[10], \*q;

ptr = f; /\* 让ptr指向函数f \*/

p = ptr(0); q = \*p; printf("%d\n", \*q);

p = ptr(1); q = \*p; printf("%d\n", \*q);

}

在上述代码中，函数有一个int型的形参。要是④，其意义与①相同，上述代码应写成“int \*ptr[10];” ，即ptr是一个有10个元素的数组，数组元素是指向整型变量的指针。

79．计算平方数时，不可能引起二义性的宏定义是

① #define SQR(x) x\*x ② #define SQR(x) (x)\*(x)

③ #define SQR(x) (x\*x) ④ #define SQR(x) ((x)\*(x))

【解】宏调用是字符列的替换，一个宏调用被替换后，替换后的字符序列就与它前后的字符列一起构成新的源程序。常有这样的情况，因宏定义对宏参数的独立性注意不够，替换后的字符列会与其前后字符列结合，产生意想不多的结果；另外，因宏调用本身独立性注意不够，宏展开的内容会被分裂，部分内容会与其前或后的其它部分先结合。如一个宏定义有可能发生上述情况，它就是一个有可能引起二义性的宏定义。对于①，若希望描述计算u+v的平方，用宏调用SQR(u+v)，产生的代码将是u+v\*u+v。这是因宏定义对参数的独立性注意不够，产生不是希望的展开代码。对于②，若希望描述计算1.0/x2，用宏调用1.0/SQR(x)，产生的代码将是1.0/(x)\*(x)，这是因宏调用本身独立性注意不够而引起的。对于③，与①的情况一样，宏调用SQR(u+v)，不能正确描述u+v的平方。对于④，在宏定义中，给参数x加上括号能保证参数的独立性，定义给全部内容也加上括号，这也保证了整个宏调用的独立性。所以，④是计算平方数时，不可能引起二义性的宏定义。所以解答为④。

80．执行下列程序

#define M(a, b) (a) > (b)?(a):(b)

main()

{ int i = 10, j = 15; printf(“%d\n”, 10\*M(i, j)); }

后输出的结果是

① 10 ② 15 ③ 100 ④ 150

【解】上述程序输出表达式10\*M(i, j)的值，该表达式中有宏调用，宏展开后的表达式为

10 \*（i）> (j) ? (i) : (j)

该表达式是一个条件表达式，它的求值顺序如下：先求10 \*（i）>(j)，因变量i和j的值分别为10和15，10\*10大于15，结果为1。因条件表达式中的条件子表达式为1，以表达式(i)的结果为条件表达式的值。(i)的值是10，条件表达式的结果亦为10。最后，程序输出10。所以解答为①。

81．设有定义语句“struct { int x; int y; } d[2] = {{1, 3}, {2, 7}};”，则

printf(“%d\n”, d[0].y/d[0].x\*d[1].x);

的输出是○。

① 0 ② 1 ③ 3 ④ 6

【解】结构数组d有两个结构元素，每个结构又由两个整型成分组成，初始化后，d[0].x = 1,d[0].y = 3, d[1].x = 2。所以表达式d[0].y /d[0].x\*d[1].x的值是6。所以正确解答是④。

82．设有定义语句“enum term { my, your = 4, his, her = his+10};”，则

printf(“%d, %d, %d, %d\n”, my, your, his, her);

的输出是○。

① 0, 1, 2, 3 ② 0, 4, 0, 10 ③ 0, 4, 5, 15 ④ 1, 4, 5, 15

【解】按枚举类型定义的规定，枚举类型team的枚举常量对应的整数分别是my=0、your=4、his=5、her=15。所以语句输出0，4，5，15。解答是③。

83．若有如下定义，则printf(“%d\n”, sizeof(them));的输出是○。

typedef union {long x[2]; int y[4]; char z[8]; } MYTYPE;

MYTYPE them;

① 32 ② 16 ③ 8 ④ 24

【解】共用型变量的成分是覆盖存储，为共用型变量分配存储时，以其中需要最多存储字节的成分进行分配。设long型数据占4个字节、int型数据占2个字节。共用型MYTYPE有三个成分，第一个成分是有两个long型元素的数组，它需要8个字节；第二个成分是有4个int型元素的数组，它也需要8个字节；第三个成分是有8个字符元素的数组，也是8个字节。所以MYTYPE类型的共用型变量只需要8个字节。正确解答是③。

1. 根据下面的定义，能打印出字母M的语句是○。

struct person {char name[9]; int age; };

struct person c[10]={{“John”, 17},

{“Paul”,19},{“Mary”,18},{“Adam”,16}};

1. printf(“%c”, c[3].name);
2. printf(“%c”, c[3].name[1]);
3. printf(“%c”, c[2].name[1]);
4. printf(“%c”, c[2].name[0]);

【解】结构类型struct person有两个成分：九个元素的字符数组name和一个基本整型age。元素类型为struct person的数组c有10个元素，其中前四个元素被设定了初值。在已设定初值的元素中，只有c[2].mane[0]的值是字符M。其余元素及元素的成分都不是字符M。解答是④。

1. 设有如下定义，则对data中的a成员的正确引用是○。

struct sk {int a; float b; } data, \*p = &data;

① (\*p).data.a ② (\*p).a ③ p->data.a ④ p.data.a

【解】结构变量data有两个成分a和b。结构指针变量p指向结构变量data。引用结构变量data的成分a有两种方法：一是直接由变量data引用其自己的成分a，写成data.a。二是通过指针变量p引用它所指的结构的成分a。这又有两种写法，分别是p->a和（\*p）.a。解答是②。

1. 设有如下定义，则对pp中的name成员的正确输入是○。

struct s {char \*name; int no; } pp;

1. scanf(“%s”, name);
2. scanf(“%s”, pp.name);
3. scanf(“%s”, &&pp.name);
4. { char buf[120];

scanf(“%s”, buf);

pp.name = (char \*)malloc(strlen(buf)+1);

strcpy(pp.name, buf);

}

【解】字符串最终存于某个字符数组中，而不是存于字符指针中。字符指针只是指向存放这个字符串的字符数组的首字符位置。通常对于结构变量中包含字符指针的情况，要为它输入字符串，得先输入到一个较大的字符数组中暂存，然后按输入字符串的长度向系统申请存储字符串的空间，并让结构的字符指针指向这个动态空间，最后将暂存于字符数组中的字符串拷贝到这个动态空间。所以解答是④。

89．设typedef union { long x[2]; short y[4][5]; char z[10];} TYPE1;

TYPE1 u1;

则printf(“%d\n”, sizeof(u1)); 的输出是

① 8 ② 10 ③ 40 ④ 58

【解】上述代码定义共用型TYPE1，并定义TYPE1类型的变量u1。共用型如同结构型一样，可以有多个不同类型的成员，但共用型变量成员的内存分配方法与结构变量成员的内存分配方法不同。结构变量的成员顺序分配内存，各成员独立占用内存。一般来说，结构变量所占用的内存字节总数是它的各成员所占内存字节数之和；共用型变量的成员重叠占用内存，共用型变量占用的字节数由其中需要最多字节数的那个成员确定。计算具体变量实际将占用多少字节，依赖于各种基本数据类型所需字节数。以一个long型数据占用4个字节、short型数据占用2个字节、char型数据占用1个字节为标准计算。对于共用型TYPE1来说，成员x需8个字节，成员y需40个字节，成员z需10个字节。这样共用型变量u1就需40个字节。输出共用型变量u1占用的字节数应是40。所以解答为③。

92．以只写方式打开一个二进制文件，应选择的使用方式是

①”a+” ②”w+” ③”rb” ④”wb”

【解】程序使用文件可以有多种不同方式。在C系统中，使用方式的不同由文件打开方式指定。首先是文件信息的存储形式，有以内存表示形式存储的二进制文件、和以字符形式存储的文本文件。要特别指定为二进制形式存储，在打开方式中要有字符b。其次是文件的读写方式。其中只读一个已存在的文件，在打开方式中用字符r；若只写建立一个新文件，或只写重建一个文件，在打开方式中用字符w。向已存在的文件只写追加信息，在打开方式中用字符a。若有读有写方式使用文件，在打开方式中有字符+。特别地，对已存在文件的有读有写，用双字符r+；在文件尾部追加信息的有读有写，用双字符a+；新建，或重建文件的有读有写，用双字符w+。由以上说明知，以只写方式打开一个二进制文件，在打开方式中有字符b和有字符w。所以解答为④。

93．若文件型指针fp已指向某文件的末尾,则函数feof(fp)的返回值是

① 0 ② -1 ③ 非零值 ④ NULL

【解】当调用fopen函数时，系统将为实参文件设定一个存放文件读写控制信息的控制块。该控制块记录有关文件持征、使用方式，当前读写位置等文件状态，并将该控制块的指针返回，作为文件指针变量的值。以后程序通过该指针调用函数时，系统能访问到这个控制块，利用该文件控制块中的控制信息，正确完成文件的读或写。所以文件指针并不真正指向文件当前读写位置，但通过文件指针，系统能在文件控制块中访问文件的当前读写位置。若某文件与文件指针fp相关联着，该文件的当前读写位置在文件的末尾时，函数调用feof(fp)的返回值是一个非零值。所以正确解答是③。

1. 下列语句中，将c定义为文件型指针的是

① FILE c; ② FILE \*c; ③ file c; ④ file \*c;

【解】如上题所述，文件控制块的类型已由系统命名为FILE。所以定义文件指针变量是c，就得用代码：“FILE \*c；”。所以正确解答是②。

1. 标准库函数fputs(p1, p2)的功能是
2. 从p1指向的文件中读一个字符串存入p2指向的内存
3. 从p2指向的文件中读一个字符串存入p1指向的内存
4. 从p1指向的内存中的一个字符串输出到p2指向的文件

④ 从p2指向的内存中的一个字符串输出到p1指向的文件

【解】标准函数调用fputs(p1, p2)完成的功能是将由字符指针p1所指的字符串内容输出到与文件指针p2相关联的文件中，即将从p1所指的内存中的字符串内容复制存入到p2所指向的文件中。所以正确解答是③。

99．以下程序中的for循环执行的次数是

#include <stdio.h>

#define N 2

#define M N+1

#define NUM (M+1)\*M/2

main()

{ int i, n = 0;

for(i = 1; i <=NUM;i++) { n++; printf(“%d”, n); }

printf(“\n”);

}

① 5 ② 6 ③ 8 ④ 9

【解】代码NUM被替换成（2+1+1）\* 2+1/2，所以其值是8。所以程序中的for循环执行的次数是8次。正确解答是③。

100．设有结构类型：

typedef struct ele {

int no; char name[20];

} Ele;

另有以下变量定义：

Ele a[20], x;

以下语句希望实现向与文件指针变量fp关联的文件输出一个结构信息，错误的代码是。

1. fwrite(a, sizeof(Ele), 1, fp);
2. fwrite(&a[2], sizeof(Ele), 1, fp);
3. fwrite(&x, sizeof(Ele), 1, fp);
4. fwrite(x, sizeof(Ele), 1, fp);

【解】函数fwrite(char \*buf, int size, int count, FILE \*fp)实现将从地址buf开始存储的，每块信息有size个字节，连续的count块信息复制存入与文件指针fp关联的文件中。这里要求调用时的第一个实参是指针。在①中因a是数组，单一的a是数组a的首元素的指针，所以是正确的。②的&a[2]是数组元素a[2]的指针，也是正确的。③中的&x是取结构变量x的指针，所以也是正确的。④中的x应表示对结构变量x的值的引用，不是指针，所以是错误的。正确解答是④。

**填充题**

1. 转义字符是由▁▁▁▁符号开始，后接单个字符或若干字符组成。

【解】在C语言中，转义字符由反斜杠字符开始，后接单个字符或若干个字符组成。

1. 在函数内，说明变量时，若省略存储类型符，系统默认其为▁▁▁▁存储类别，该存储类别的类型符为：▁▁▁▁。

【解】在函数内，说明变量时，若省略存储类型符，系统就默认其为自动存储类别，该存储类型用auto标识。

1. 设有下列运算符:<<、+、++、&&、<=，其中优先级最高的是▁▁▁▁，优先级最低的是▁▁▁▁。

【解】对运算符<<、+、++、&&、<=，按它们的优先级自高到低的顺序排列为：

++、+、<<、<=、&&

所以，优先级最高的是++，优先级最低的是&&。

1. 设二进制数A是00101101，若想通过异或运算A^B使A的高4位取反，低4位不变，则二进制数B应是▁▁▁▁。

【解】按位加运算的一个重要应用是让某个整型变量的二进位位串信息的某些位信息反向，0变成1，而1变成0。这只要设计这样一个位串信息，让要变反的位为1，不要改变的位为0，用这个位串信息与整型变量按位加就能得到希望的结果。要使字节的高4位取反，低4位不变，则需要位串信息是11110000，写成八进制数是0360，写成十六进制数为0xF0。

1. 设a=3，b=2，c=1，则a>b的值为▁▁▁▁，a>b>c的值为▁▁▁▁。

【解】因a的值为3，b的值是2，条件a>b为真，其值为1。表达式a>b>c的求值顺序是计算a>b，结果为1，接着计算1>c，因c的值为1，条件1>c为假，结果为0。

1. 能表述“20<x<30或x<-100”的C语言表达式是▁▁▁▁。

【解】首先表述20<x<30的C表达式可写成20<x && x < 30。所以表述“20<x<30或x<-100”的C表达式为20<x && x < 30 || x < -100。

1. 结构化程序设计规定的三种基本结构是▁▁▁▁结构，选择结构和▁▁▁▁结构。

【解】结构化程序设计的三种基本控制结构是顺序结构、选择结构和循环结构。

。

23．以下两条if语句可合并成一条if语句为▁▁▁▁。

if (a <= b) x = 1;

else y = 2;

if(a > b) printf(“\*\*\*\*y = %d\n”, y);

else printf(“####x = %d\n”, x);

【解】在以上两条if语句中，两个条件刚巧相反。若将前一个if语句的第一个成分语句与第二个if语句的第二个成分语句合并；第一个if语句的第二个成分语句与第二个if语句的第一个成分语句合并，写成一条if语句如下：

if (a <= b) {x = 1; printf(“####x = %d\n”, x); }

else { y = 2; printf(“\*\*\*\*y = %d\n”, y);}

24．设i，j，k均为int型变量，则执行完下面的for语句后，k的值为▁▁▁▁。

for(i = 0, j = 10; i <= j; i++, j--) k = i+j;

【解】该fot语句以i为0、j为10初始化，循环条件是i <= j，每次循环后i增1、j减1，循环体是将i与j的和赋给k。这样变量k将保存的是最后一次赋给它的值。一次循环后i为1、j为9，二次循环后i为2、j为8，……，五次循环后i为5、j为5，继续第六次循环，将i与j的和10存于k后，i为6、j为4，结束循环。所以循环执行后k为10。

25．下列程序的功能是输入一个整数，判断是否是素数，若为素数输出1，否则输出0，请为程序填空。

main()

{ int i, x, y = 1;

scanf(“%d”, &x);

for(i = 2; i<=x/2; i++)

if ▁▁▁▁ { y = 0; break;}

printf(“%d\n”, y);

}

【解】为判数x是否是素数，预置变量y的值为1（x可能是素数），然后用循环，用2至x/2的整数i测试对x的整除性，若能被其中的某个整数整除，则x是合数，置y的值为0（x不是素数），结束测试循环。若都不能整除，则x为素数。i能否整除x，可用求余运算x%i等于0来判定。所以程序中的空框可填x%i == 0。测试x为素数的代码可写成：

y = 1;

for(i=2; i <=x/2; i++)

if(x%i == 0) { y = 0; break; }

1. 根据变量定义“static int b[5], a[][3]={1,2,3,4,5,6}；”，b[4]=▁▁▁▁▁，a[1][2]= ▁▁▁▁▁。

【解】系统规定，静态变量定义时，若未初始化，系统自动给定二进制代码全0的值。所以静态数组b的元素全为0，当然包括b[4]也为0。静态数组a是两维数组，每行有三个元素，在初始化中给出了六个初值，所以数组a有2行。由于数组的元素按行顺序存储，并按行顺序初始化。前三个值是第一行，后三个值是第二行。a[1][2]是第二行的第三列元素，也是a的最后一个元素，所以其值是6。

1. 下面程序的功能是输出数组s中最大元素的下标，请填空。

main()

{ int k, p，s[ ] = {1, -9, 7, 2, -10, 3};

for(p = 0, k = p; p < 6; p++)

if(s[p] > s[k]) ▁▁▁▁▁

printf(“%d\n”, k);

}

【解】为要寻找数组中的最大元素的下标，需先预设1个临时最大元素的下标，并顺序逐一考察数组的元素，当发现当前元素比临时最大元素更大时，就用当前元素的下标更新临时最大元素下标。直至考察了数组的全部元素后，这临时最大元素下标就是数组的最大元素下标。通常预设的最大元素下标是数组的首元素下标，考察是从首元素开始顺序向后继元素考察。程序中，存储临时最大元素下标的变量是k，变量p控制顺序考察的循环控制变量。当发现当前元素s[p]比临时最大元素s[k]更大时，应该用p更新k。所以在空框处应填入代码“k=p；”。

1. 下面程序的功能是将一个字符串str的内容颠倒过来，如“1234567”变换成“7654321”

#include <string.h>

main()

{ int i, j; ▁▁▁▁▁ ; char str[ ] = {“1234567”};

for(i = 0, j = strlen(str) ▁▁▁▁▁ ; i < j; i++, j--)

{ k = str[i]; str[i] = str[j]; str[j] = k;}

printf(“%s\n”, str);

}

【解】颠倒一个字符串中的字符，就是首尾对应的元素两两交换。简单地可用两个游标变量i和j，i是前端元素的下标，j是后端元素的下标，交换以这两个变量值为下标的元素str[i]和str[j]。开始时，i的值为0，j的值为字符串末元素的下标（字符串长度减1）。每次交换后，i增1，j减1。继续交换的条件是str[i]还位于str[j]的前面，即i<j。字符串末元素的下标是它的长度减1，所以在第二个空框处应填入-1。程序为了交换str[i]和str[j]，使用了变量k，该变量应在程序的变量定义部分中一起定义，所以在第一个空框处应填入k。

1. 以下程序可把输入的十进制数以十六进制数的形式输出，请填空。

main()

{ char b[17] = {“0123456789ABCDEF”};

int c[64], d, i = 0, base = 16;

long n;

pointf(“Enter a number:\n”); scanf(“%ld”, &n);

do { c[i] = ▁▁▁▁▁ ; i++; n = n/base; }

while (n != 0);

printf(“Transmite new base:\n”);

for(--i; i >= 0; --i)

{ d = c[i]; printf(“%c”, b▁▁▁▁▁); }

printf(“\n”);

}

【解】程序中，字符数组b存储十六进制的16个数字符，整型数组c存储译出的十六进制数的各位数值。从整数n译出它的各位十六进制数值，需采用除16取余的方法，即求n除16的余，得到它的十六进制的个位数，接着将n除以16。在n不等于0的情况下循环，能顺序求出n的十六进制的个位数、十位数、百位数等等。程序中变量base已预置16，所以在第一个空框处可填代码n%base或n%16。当n的十六进制数的个位、十位、百位等依次存放于数组c中后，就从最高位至最低位，参照数组c[i]的内容d（以其内容为下标），取十六进制数字符表中的字符b[d]输出。所以在第二个空框处可填入代码[d]。

32．“\*”称为▁▁▁▁▁运算符，“&”称为▁▁▁▁▁运算符。

【解】单目运算符“\*”称为取内容运算符，单目运算符“&”称为取地址运算符。

33．若两个指针变量指向同一个数组的不同元素，可以进行减法运算和▁▁▁▁▁运算。

【解】若两个指针变量指向同一个数组的不同元素，可以进行减法运算求它们所指元素相差多少元素。进行关系运算，判定它们所指元素的前后、或是否指向同一个元素等。

1. 设有char \*a=”ABCD”，则printf(“%s”, a)的输出是▁▁▁▁▁；而printf(“%c”, \*a)的输出是▁▁▁▁▁。

【解】若给字符指针变量a赋一个常量字符串”ABCD”，实际上是给a赋指向字符串常量首字符’A’的指针。程序通过它访问字符串中的各字符。如能用代码printf(“%s”, a)输出这个字符串常量”ABCD"的字符列ABCD，用代码printf(“%c”, \*a)输出a所指的字符A。

1. 以下程序的功能是从键盘上输入若干个字符（以回车符作为结束）组成一个字符串存入一个字符数组，然后输出该字符数组中的字符串。请填空。

#include <ctype.h>

#include <stdio.h>

main()

{ char str[81], \*sptr;

int i;

for(i = 0; i < 80; i++)

{ str[i] = getchar(); if (str[i] == ‘\n’) break; }

str[i] = ▁▁▁▁ ; sptr = str;

while(\*sptr) putchar(\*sptr▁▁▁▁);

}

【解】从键盘输入字符行，通过循环逐一输入字符，当发现输入字符是换行符时，结束循环。为了使输入的字符列变成字符串，必须在输入字符列之后，原存储换行符的元素改为存储字符串的结束标记符，需用代码str[i] = ‘\0’，所以在第一框填入代码‘\0’。当要将存于字符数组str中的字符串通过循环逐一输出字符串的字符时，可以用字符指针sptr，让sptr遍历整个字符串，每次循环时，输出sptr所指字符，并让sptr增1，即可用代码\*sptr++实现，所以在第二框填入代码++。

1. 若自定义函数要求返回一个值，则应在该函数体中有一条▁▁▁▁▁语句；若自定义函数要求不返回一个值，则应在该函数说明时加一个类型符▁▁▁▁▁。

【解】若自定义的函数有返回值，则函数返回时应执行带表达式的return语句返回，该表达式的值将作为函数调用的结果。为了强调函数是一个不返回值的函数，应在该函数说明时，加上一个void类型说明符。

1. 若给fun函数的形参s传送字符串：”∪∪6354abcc” （其中∪表示空格字符），则函数的返回值是▁▁▁▁▁。

#include <ctype.h>

long fun(char s[])

{ long n; int sign;

for(; isspace(\*s); s++);

sign = (\*s ==’-‘) ? –1 : 1;

if (\*s == ‘+’ || \*s == ‘-‘) s++;

for(n = 0; isdigit(\*s); s++) n = 10\*n + (\*s – ‘0’);

return sign\*n;

}

【解】函数fun的第一个for循环跳过字符串的前导空白符，接着分析第一个非空白符是否是负号，若是置变量sign为-1；否则，置变量sign为1。接着的if语句在第一个非空白字符是负号或正号时，跳过该字符。以后的for循环将后面的数字符当作十进制数的各位数字译出一个长整数n。遇字符a时，结束译数循环。最后，函数返回sign与n的乘积。所以若调用函数时提供的实参是” ∪∪6354abc”，函数返回值是6354。

1. 下面函数要求计算两个整数x、y之和，并通过形参z传回该和值，请填空。

void add(int x, int y, ▁▁▁▁▁ z)

{ ▁▁▁▁▁ = x+y; return; }

【解】函数希望通过形参z带回结果，形参z必须是指针类型的，由于返回整型结果，所以z的说明是int \* z。函数返回前必须先把结果通过形参z间接引用(写成\*z)传回。

1. 函数my\_cmp()的功能是比较字符串s和t的大小，当s等于t时返回0，否则返回s和t的第一个不同字符的ASCII码差值，即s > t时返回正值，当s < t时返回负值。请填空。

my\_cmp(char \*s, char \*t)

{ while (\*s == \*t)

{ if (\*s == ‘\0’) return 0;

++s; ++t;

}

return ▁▁▁▁▁;

}

【解】两字符串大小比较必须从它们的首字符开始，在对应字符相等情况下循环，直至不相等结束。相等时，若字符串已到了字符串的结束标记符，则两字符串相同，函数返回0值；如还有后继字符，则准备比较下一对字符。对应字符不相同，循环结束。循环结束时，就以两个当前字符的差返回。所以在空框处应填入\*s-\*t，保证在s > t时返回正值，当s < t时返回负值。

union aa {float x; float y; char c[6]; };

struct st{ union aa v; float w[5]; double ave; } w;

【解】共用型aa有三个成分，一个是float型成分x,一个是float型成分y，再一个是有六个元素的字符数组。若float 型数据占4个字节，则类型aa 数据要占用6个字节。结构变量w有三个成分，一个是共用型成分aa , 要占用6个字节；一个是有5个float型元素的数组w，要占用20个字节；一个是double型的ave。若double型数据要8个字节，则结构变量w要占用34个字节。

1. 在对文件进行操作的过程中，若要求文件的现行位置回到文件的开头，应当调用的函数是▁▁▁▁函数。

【解】若要让文件随机地从某个现行位置回到文件的开头，可调用库函数rewind()。

1. 下面的程序用来统计文件中字符的个数，请填空。

#include <stdio.h>

main()

{ FILE \*fp;

long num = 0;

if ((fp = fopen(“fname.dat”, “r”)) == NULL)

{ printf(“Cant’t open file!\n”); exit(0); }

while ▁▁▁▁{ fgetc(fp); num++; }

printf(“num = %d\n”, num);

fclose(fp);

}

【解】若要统计文件中的字符个数，程序设计一个计数器，并预置该计数器为0值，并打开文件。在正确打开情况下，接着是一个循环。循环在文件未结束的情况下，反复读入字符，并增加字符计数器。所以循环条件可写成：!feof(fp)。程序有一个错误，由于num变量的类型是long型的，输出它的值应在格式符d之前加上格式修饰符‘l’， 不加这个修饰符只有在long型与int型实现相同的系统才是正确的。另外，要统计文件的长度，最好用二进制方式打开，通过读入的字节数求出文件的长。这是因为正文文件方式打开时，文件中的双字符（回车字符和换行字符）读入后将被当作一个字符存储，二进制方式不进行这样的转换。

54．设有定义“#define F(N) 2\*N”，则表达式F（2+3）的值是▁▁▁▁▁▁。

【解】F(N)被定义成2\*N，表达式F(2+3)被替换成2\*2+3，其值是7。

1. 设有下列程序：

#include <stdio.h>

main(int argc, int \*argv[ ])

{ while(--argc > 0) printf(“%s”, argv[argc]);

printf(“\n”);

}

假定上述程序编译连接成目标程序名为p.exe，如果输入如下的命令：

p 123 AAA BBB↙ （其中的↙表示回车）

则输出结果是▁▁▁▁▁▁。

【解】程序启动时，参数argc的值是4，由于循环条件是—argc > 0，是先减1后判大于0，循环共进行三次，顺序输出argv[3]，argv[2]和argv[1]，即输出BBBAAA123。

56．以下分别是算术、位、关系和逻辑运算符

+， &， <, !

其优先级最低的是\_\_\_\_\_\_\_\_\_ 。

【解】C语言的运算符十分丰富，并且运算符的优先级也很多。各运算符的优先级大致是按运算符的意义分类的，优先级自高到低分别是：

圆括号、数组元素运算符[]、结构或共用体成员运算符.和->

单目运算符（如否定，求反、自增自减、取地址、取内容、取正、取负、数据长度）

算术运算符（先乘、除、求余，后加、减）

移位运算符

关系运算符（先大小，后相等和不等）

双目位运算符（依次为按位与、按位加、按位或）

双目逻辑运算符（依次为逻辑与、逻辑或）

三目条件运算符

赋值运算符（先简单赋值，后自反赋值）

逗号运算符

本题给出的运算符+、&、<、! 分别是算术运算符的加、双目逻辑运算符的逻辑与、关系运算符的小于和单目运算符的逻辑否定。在这4种运算符中，优先级最低的是双目逻辑运算符的逻辑与&。所以解答是&。

58．设int a, b = 10;执行a = b << 2 + 1；后a的值是\_\_\_\_\_\_\_\_\_ 。

【解】用圆括号表示上述表达式的计算顺序，可写成

a = (b << (2+1))

即将b的值左移3位后存于变量a。一般地，在左移没有使数值二进位丢失的情况下，每左移一位，相当于乘2。这样，10左移3位后的值是80，上述表达式使a的值为80。所以解答是80。

62．在C语言中，\_\_\_\_\_\_\_\_\_ 类型变量的所有成员以覆盖方式共享存储单元。

【解】在C语言中，只有共用型变量的所有成员以覆盖方式共享存储单元。共用型变量也称为共用体、联合体，或union类型变量。

65．设 int a[3][4], \*p = a[1]; 该指针变量p指向的元素是\_\_\_\_\_\_\_\_\_。

【解】在C语言中，一个二维数组可以看作元素是一维数组的一维数组。如问题所给的二维数组a，可以看作由3个元素a[0]、a[1]、a[2]组成的一维数组，其中a[0]、a[1]、a[2]又各自是由4个元素组成的一维数组。如a[1]表示由a[1][0]、a[1][1]、 a[1][2]、a[1][3]这4个元素组成的数组。按这一观点，表达式a[1]就是这个一维数组首元素a[1][0]的指针。所以，问题中的指针变量p指向的是a[1][0]。按数组元素的不同标记形式，a[1][0]也可写成\*(\*(a+1)+0)，或简写成\*\*(a+1)。

67．将一字符串中的小写英文字母改成大写。

main()

{ int i = 0; char str[80];

scanf(“%s”, str);

while (\_\_\_\_\_\_)

{ if(\_\_\_\_\_\_) str[i] -= ‘a’ – ‘A’;

\_\_\_\_\_\_ ;

}

printf(“%s\n”, str);

}

【解】为了改写字符串中的小写英文字母，需要顺序地考察字符串的每一个字符。当发现当前元素str[i]是小写英文字母时，就改写str[i]，减去小写字母与对应大写字母ASCII代码的差。由于要顺序考察字符串的每个字符，所以框1的循环条件是当前字符str[i]不是字符串结束符，写成str[i] != ‘\0’或简写成str[i]。仅当str[i]是小写英文字母时才要改写，表示str[i]是小写英文字母的条件框2可写成str[i]>=’a’ && str[i]<=’z’。每考察了一个字符后，就要准备考察下一个字符，框3处的语句是让变量i增1，可写成i++，或++i，或i=i+1，或i+=1等。

73．以下程序的输出结果是\_\_\_\_\_\_。

main()

{ static char a[ ] = {‘\*’, ‘\*’, ‘\*’, ‘\*’, ‘\*’};

int i, j, k;

for(i = 0; i < 5; i++)

{ printf(“\n”);

for(j = 0; j < i; j++) printf(“%c”, ‘ ‘);

for(j = 0; j < 5; j++) printf(“%c”, a[k]);

}

printf(“\n”);

}

【解】程序的外循环控制循环5次，每次开始先输出一个换行符，共输出5行信息。对于i（自0开始计算）行，先输出i个空白符，接着输出字符数组a中的5个字符，由于a中的5个字符都是字符\*，所以输出5个\*字符。这样程序是输出以下形式的图案：

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

1. 以下程序的输出结果是\_\_\_\_\_\_。

main()

{ char \*a[6] = {“AB”, “CD”, “EF”, “GH”, “U”, “KL”};

int i;

for(i = 0; i < 4; i++) printf(“%s”, a[i]);

printf(“\n”);

}

【解】程序定义了有6个元素的字符指针数组a，数组a的各元素指向字符串常量。程序的for循环遍历了a的前4个元素，用字符串输出格式，输出指针数组a前4个元素所指的字符串。所以程序输出：ABCDEFGH。

1. 以下程序的功能是\_\_\_\_\_\_。

main()

{ int i, a[10], \*p = &a[9];

for(i = 0; i < 10; i++) scanf(“%d”, &a[i]);

for(; p>=a; p--) printf(“%d\n”, \*p);

}

【解】程序定义了有10个元素的整型数组a，另定义指针变量p，并让它指向a的最后一个元素a[9]。执行代码的第一个循环是顺序输入数组a的10个元素。第二个循环利用指针p逆序遍历数组a，将数组a的各元素逆序输出。所以程序的功能是输入10个整数，并逆序输出这10个整数。

1. 以下程序的输出结果是\_\_\_\_\_\_。

main()

{ char str[ ] = {‘A’, ‘B’, ‘C’, ‘\0’, ‘D’, ’E’, ‘F’, ‘\0’}, \*p = str;

for(;p-str<sizeof(str); p++)

printf(“%s\n”, p);

}

【解】程序用字符指针，顺序遍历字符数组，该字符数组中存放着两个字符串。当指针所指的字符还在第一个字符串中时，输出第一个字符串中的部分内容。当指针所指的字符是第二个字符串中的字符时，输出第二个字符串中的部分内容。所以程序输出：

ABC

BC

C

DEF

EF

F

1. 以下程序的输出结果是\_\_\_\_\_\_。

struct stu

{ int num; char \*name; int score; };

struct stu \* py(struct stu \*p)

{ int s = p;

for(p++; p->no>0; p++)

if(s->score < p->score) s = p;

return s;

}

main()

{ struct stu student[ ] = {{1001, “Sun”, 76}, {1002, “Ling”, 88},

{1003, “Shen”, 76}, {0, “ “, 0}};

struct stu \*p; p = py(student);

printf(“NO:%d NAME:%s SCORE:%d\n”, p->num, p->name, p->score);

}

【解】函数py的功能是在给定的结构数组中找出成绩最高的结构元素，并返回该结构的指针。主函数利用该指针输出这个结构的内容。所以程序输出：

NO：1002 NAME:Ling SCORE:88

1. 设下面的程序经编译连接后生成的目标程序名为：ABC.EXE，写出输入以下命令行后的输出结果是\_\_\_\_\_\_。

ABC FILE1 FILE2↙ （其中的↙表示回车）

#include <stdio.h>

main(int argc, char \*argv[ ])

{ int i;

printf(“%d\n”, argc);

for(i = 0; i < argc; i++) printf(“%s “, argv[i]);

printf(“\n”);

}

【解】程序首先输出参数argc的值3，然后循环3次，顺序输出ABC、FILE1和FILE2，并回车：

3

ABC FILE1 FILE2

1. 以下程序的功能是\_\_\_\_\_\_。（字符0的ASCII码为16进制的30）

main()

{ union{ char c; char i[4]; }z;

z.i[0] = 0x39; z.i[1] = 0x36;

printf(“%c\n”, z.c);

}

【解】共用型变量z有两个成分，一个是字符型的c，另一个是有4个字符的数组i。由于共用型的成分是复盖存储，成分c的内容也是成分i的首元素。程序给z的i成分的首元素置十六进制代码0x39，即十进制数57。数57是字符’9’的ASCII代码。用字符输出格式，输出该变量的值，将输出字符’9’。

1. 以下程序的输出结果是\_\_\_\_\_\_。

main()

{ enum em{ em1 = 3, em2 = 1, em3};

char \*aa[ ] = {“AA”, “BB”, “CC”, “DD”};

printf(“%s%s%s\n”, aa[em1], aa[em2], aa[em3]);

}

【解】由枚举类型em的定义知，枚举常量em1的值是3，枚举常量em2的值是1，枚举常量em3的值是2。所以程序运行后输出：

DD BB CC

1. 假定在当前盘当前目录下有2个文本文件，其名称和内容如下：

文件名 a1.txt a2.txt

内容 121314# 252627#

以下程序的输出结果是\_\_\_\_\_\_。

#include <stdio.h>

main()

{ FILE \*fp;

if((fp = fopen(“al.txt”, “r”)) == NULL)

{ printf(“Can not open file!\n”);

exit(1);

}

else { fc(fp); fclose(fp); }

if((fp = fopen(“a2.txt”, “r”)) == NULL)

{ printf(“Can not open file!\n”);

exit(1);

}

else {fc(fp); fclose(fp); }

}

void fc(fpl)

FILE \*fp1;

{ char c;

while ((c = fgetc(fp1)) != ‘#’) putchar(c);

}

【解】程序以读方式打开文件a1.txt，若能正常打开，则调用函数fc。接着同样地，以读方式打开文件a2.txt，若能正常打开，也调用函数fc。函数fc从指定的文件读入字符，在读入字符不是字符’#’情况下循环，将读入字符输出。对于文件a1.txt，程序输出121314，对于文件a2.txt，程序输出262627。所以程序输出121314252627。

1. 以下程序的输出结果是\_\_\_\_\_\_。

#define PR(ar) printf(“%d,”,ar)

main()

{ int j, a[ ] = {1, 3, 5, 7, 9, 11, 15}, \*p = a+5;

for(j = 3; j; j--)

switch(j)

{ case 1:

case 2: PR(\*p++); break;

case 3:PR(\*(--p));

}

printf(“\n”);

}

【解】程序的宏定义PR以d格式输出参数的值和逗号字符。程序给数组a赋初值，并由指针p指向元素a[5]。程序让j从3至1，共循环3次。当j是3时，让p减1，并输出p所指元素的值9；当j是2和1时，输出p所指的数组元素的值，并让p增1，顺序输出9和11。所以程序输出：

9，9，11，

1. 以下程序的输出结果是\_\_\_\_\_\_。

main()

{ char str[ ] = “1234567”; int i;

for(i = 0; i < 7; i += 3)

printf(“%s\n”, str+i);

}

【解】程序中，字符数组str中存有字符串“1234567”。循环语句受变量i控制，i的初值为0，循环条件是i < 7，每次循环后i增3。所以，i的值依次为0、3、6，共进行了3次循环。每次循环输出str中从str+i位置开始的部分字符串内容。第一次循环，i 的值为0，输出内容是“1234567”；第二次循环，i 的值为3，输出内容是“4567”；第三次循环，i 的值为6，输出内容是“ 7”。所以，程序将输出：

1234567

4567

7

97．以下程序的输出结果是\_\_\_\_\_\_。

conv (b)

int b;

{ if (b >= 2) conv(b/2);

printf(“%d”, b%2);

}

main()

{ int d;

scanf(“%d”, &d); conv(d);

}

输入22↙

【解】程序中的函数conv有一个整数形参b，该函数是一个递归函数。函数的递归算法是当形参b大于等于2时就继续以b/2 为实参递归。递归调用返回时，输出b%2。即当b是偶数时，输出0；b是奇数时，输出1。或者说，输出b的二进制表示的个位。主函数以输入值22为实参调用函数conv，使函数conv引起以下一系列的递归调用和输出：

conv(b=22) → conv(b=11) → conv(b=5) → conv(b=2) → conv(b=1)

↓ ↓ ↓ ↓ ↓

输出0 ← 输出1 ← 输出1 ← 输出0 ← 输出1

↓

主函数

由上述递归调用的执行过程知，对于输入22，程序的这次执行将输出10110。

100．以下程序的功能是\_\_\_\_\_\_\_。

#include <stdio.h>

main()

{ int a[20], i, base = 10;

int n;

printf("Enter a number!\n"); scanf("%d", &n); /\* 整数输入 \*/

i = 0; /\* 对n进行分拆，各位数字自低位到高位存于数组a \*/

do { a[i++] = n % base;

n /= base;

} while (n);

for(i--; i >= 0; i--) /\* 自高位到低位输出 \*/

printf("%d%c", a[i], i ? ',' : '\n');

}

【解】程序的主要工作是从输入整数分拆出它的十进制整数的各位数字，将分拆出来的各位数字存于数组中，然后将这些数字自高位到低位的顺序逐位输出。程序用一个循环将一个整数按十进制数的要求分拆，每次循环求出被分拆数除以10的余数的办法求出它的个位，每分拆出一位就将被分拆的数除以10。循环过程直至被分拆的数为0结束。