

# 程序设计基础 (C 语言版 )习题

第01章 绪论

## 一．选择题

1.以下叙述中正确的是 \_\_\_\_\_。

- A) 程序设计的任务就是编写程序代码并上机调试
- B) 程序设计的任务就是确定所用数据结构
- C) 程序设计的任务就是确定所用算法
- D) 以上三种说法都不完整

【解析】 程序设计通常分为问题建模、算法设计、编写代码和编译调试四个阶段。所以选项 A)、 B)、 C) 说法都不完整。

2.以下关于简单程序设计的步骤和顺序的说法中正确的是。

- A) 确定算法后，整理并写出文档，最后进行编码和上机调试
- B) 首先确定数据结构，然后确定算法，再编码，并上机调试，最后整理文档
- C) 先编码和上机调试，在编码过程中确定算法和数据结构，最后整理文档
- D) 先写好文档，再根据文档进行编码和上机调试，最后确定算法和数据结构

【解析】 设计一个能解决实际问题的计算机程序需要经过以下几个过程： 建立模型。 算法设计： 给出解决问题的步骤，即算法。 算法表达：选择一种表达算法的工具，对算法进行清晰的表达。 编写程序：选择一种程序设计语言，把以上算法程序化，这称为编写程序。 程序调试：对编写好的程序进行调试，修改程序中的错误。 程序文档编写与程序维护。综上所述， B ) 选项是符合上述描述的，其他选项不恰当。

3.叙述中正确的是 \_\_\_\_\_。

- A) C 程序的基本组成单位是语句
- B) C 程序中的每一行只能写一条语句
- C) 简单 C 语句必须以分号结束
- D) C 语句必须在一行内写完

【解析】函数是 C 程序的基本组成单位； C 语言书写风格很自由，不但一行可以写多个语句，还可以将一个语句写在多行中。故本题答案为 C)。

4.以下叙述中正确的是。

- A) C 程序中的注释只能出现在程序的开始位置和语句的后面
- B) C 程序书写格式严格，要求一行内只能写一个语句
- C) C 程序书写格式自由，一个语句可以写在多行上
- D) 用 C 语言编写的程序只能放在一个程序文件中

【解析】在 C 语言中，注释可以加在程序中的任何位置，选项 A) 错误。 C 程序可以分模块写在不同的文件中，编译时再将其组合在一起，选项 D) 错误。 C 程序的书写风格很自由，不但一行可以写多个语句，还可以将一个语句写在多行中。所以正确答案为选项 C)。

5.以下关于 C 语言的叙述中正确的是。

- A) C 语言中的注释不可以夹在变量名或关键字的中间
- B) C 语言中的变量可以在使用之前的任何位置进行定义
- C) 在 C 语言算术表达式的书写中 ,运算符两侧的运算数类型必须一致
- D) C 语言的数值常量中夹带空格不影响常量值的正确表示

【解析】 C 语言中注释可以放在任何位置，但不能夹在变量名或关键字中间，选项 A 正确； C 语言中的变量要在使用之前定义， C 标准规定定义位置在相关程序块的首部，选项 B 错误； C 语言中两侧数据类型可以不一致，系统可进行强制类型转换，选项 C 错误； C 语言数值常量中不允许存在空格，选项 D 错误。

6.以下关于 C 语言数据类型使用的叙述中错误的是。

- A) 若要准确无误地表示自然数 ,应使用整数类型

- B)若要保存带有多位小数的数据 ,应使用双精度类型
- C)若要处理如 "人员信息 "等含有不同类型的相关数据 ,应自定义结构体类型
- D)若只处理 "真 "和"假"两种逻辑值 ,应使用逻辑类型

【解析】 C语言中没有逻辑类型，若只处理 "真 "或 "假"两种逻辑值，可以使用整型数 "1" 或 "0"表示，故选 D。

第02章 基本数据类型

一 . 选择题

1.有以下定义： int a；long b；double x，y；则以下选项中正确的表达式是。

- A)a%(int)(x - y)      B)a = x != y；      C)(a\*y)%b      D)y = x + y = x

【解析】 A ) 选项中如果 x与y的值相等那么取余时就会有除数为 0的情况。 C ) 选项中取余的两个数据都应 为整数，不能有一方为实型变量，而 a\*y的结果为 double 型。 D ) 选项表达式本身就错误，不能给表达式赋值。所以，本题答案为 B )。

2.有以下程序

```
#include<stdio .h>

main()
{
    int s , t , A = 10 ; double B = 6 ;

    s = sizeof(A) ; t = sizeof(B) ;

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

}
```

在 VC6 平台上编译运行，程序运行后的输出结果是

- A)2,4      B)4,4      C)4,8      D)10,6

【解析】 sizeof的作用就是返回一个对象或者类型所占的内存字节数。在 VC6 中整型占 4个字节，双精度实型占 8个字节，所以选C )。

3.若有定义： double a=22;int i=0,k=18;; 则不符合 C语言规定的赋值语句是。

- A)a=a++,i++;      B)i=(a+k)<=(i+k);
- C)i=a%11;      D)i=!a;

【解析】 C语言中取余运算符两侧的操作数只能是整型 (若为char 型，则会自动转换成整型 )。

4.有以下程序：

```
#include <stdio.h>

main()
{
    int a = 1 , b = 0 ;

    printf("%d , " , b = a + b) ;

    printf("%d\n" , a = 2*b);

}
```

程序运行后的输出结果是 \_\_\_\_\_。

- A)0,0      B)1,0
- C)3,2      D)1,2

【解析】 执行第一个 printf 语句时， b = a + b = 1，所以输出 1，执行第二个 printf 语句时， a = 2\*b=2，所以输出结果为 2。

5.若有定义语句： int a=3，b=2，c=1；，以下选项中错误的赋值表达式是。

- A)a=(b=4)=3；      B)a=b=c+1；      C)a=(b=4)+c；      D)a=1+(b=c=4)；

【解析】 在 A ) 选项中，赋值语句 b=4先把 4赋值给整型变量 b，之后执行 (b=4)=3 是把常量 3赋值给常量 4，但是由于 C语言规定赋值运算符的左侧只能是变量，不能是常量或者表达式，因此 A ) 选项错误。

6.有以下定义： int a；long b；double x，y；则以下选项中正确的表达式是。

- A)a%(int)(x - y)      B)a = x != y；      C)(a\*y)%b      D)y = x + y = x

【解析】 A ) 选项中如果 x与y的值相等那么取余时就会有除数为 0的情况。 C ) 选项中取余的两个数据都应 为整数，不能有一方为实型变量，而 a\*y的结果为 double 型。 D ) 选项表达式本身就错误，不能给表达式赋值。所以，本题答案为 B )。

7.若有定义语句： `int x = 10 ;` , 则表达式 `x - = x + x`的值为 \_\_\_\_\_。

- A) - 20                      B) - 10  
C)0                          D)10

【解析】 `x - = x + x`可化为：`x = x - (x + x)`，由于x初始值为 10，所以计算后， `x = 10 - 20 = - 10`，因此选 B)。

## 二．填空题

1.以下程序的功能是：将值为三位正整数的变量 `x`中的数值按照个位、十位、百位的顺序拆分并输出。请填写。

```
#include<stdio.h>

main()
{int x=256;

    printf("%d-%d-%d\n",__,x/10%10,x/100);

}
```

参考答案： `x%10`

【解析】 `x=256`，`x%10=6`。

2.以下程序运行后的输出结果是 \_\_\_\_\_。

```
#include<stdio.h>

main()
{   int a ;

    a=(int)((double)(3/2)+0.5+(int)1.99*2)    ;

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

}
```

参考答案： 3

【解析】本题中 `3/2=1`，`(double)(3/2)=1`，`(int)1.99*2=1*2=2`，因此 `(int)(1+0.5+2)=(int)3.5=3`。

3.若有定义语句： `int a = 5 ;` , 则表达式：`a + +`的值是 \_\_\_\_\_。

参考答案： 5

【解析】 `a + +`的含义是在使用 `a` 值以后，使 `a` 值加 1，所以 `a + +`的值为 5。

4.若有语句 `double x = 17 ; int y ;` , 当执行 `y = (int)(x/5)%2` ；之后 `y`的值为 \_\_\_\_\_。

参考答案： 1

【解析】 `y = (int)(x/5)%2 = (int)(3.4)%2 = 3%2 = 1`。

5.设变量 `a`和`b`已正确定义并赋初值。请写出与 `a - = a + b`等价的赋值表达式 \_\_\_\_\_。

参考答案： `a=-b`

【解析】 `a - = a + b` 等价于 `a = a - (a + b)`，等价于 `a = - b`。

## 第03章                      顺序结构

### 一.选择题

1.有以下程序

```
#include<stdio.h>

main()

{char a,b,c,d;

    scanf("%c%c",&a,&b);

    c=getchar();d=getchar();

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

}
```

当执行程序时 ,按下列方式输入数据 (从第 1列开始 ,<CR> 代表回车 ,注意 ,回车也是一个字符 )

12<CR>

34<CR>

则输出结果是

- A)1234
- B)12
- C)12
- D)12
- 3
- 34

【解析】程序根据用户输入分别给字符型变量 a、b、c、d赋值为 '1'、'2'、'<CR>'、'3'，因此输出到屏幕得到选项 C 中的格式。

2.有以下程序段：

```
char ch ;  
  
int k ;  
  
ch = a ;  
  
k = 12 ;  
  
printf("%c , %d , " , ch , ch , k) ;  
  
printf("k = %d\n" , k) ;
```

已知字符 a 的 ASCII 码值为 97，则执行上述程序段后输出结果是

- A) 因变量类型与格式描述符的类型不匹配输出无定值
- B) 输出项与格式描述符个数不符，输出为零或不定值
- C)a,97,12k = 12
- D)a,97 , k = 12

【解析】输出格式控制符 %c 表示将变量以字符的形式输出；输出格式控制符 %d 表示将变量以带符号的十进制整型数输出。

所以第一个输出语句输出的结果为 a,97，第二个输出语句输出的结果为 k= 12，所以选项 D) 为正确答案。

3.有以下程序

```
#include<stdio.h>  
  
main()  
{ char a [ 30 ], b [ 30 ] ;  
  
scanf("%s" , a) ;  
  
gets(b) ;  
  
printf("%s\n%s\n" , a,b) ;  
  
}
```

程序运行时若输入：

how are you? I am fine < 回车 >

则输出结果是

- A)how are you?  
I am fine
- B)how  
are you? I am fine
- C)how are you? I am fine
- D)how are you?

【解析】scanf 函数会把空格字符作为输入的字符串之间的分隔符，因此字符串 a 为 "how"。gets() 函数读取字符串时，直至接受到换行符或 EOF 时才停止，并将读取的结果存放在指针所指向的字符数组中，因此其余输入的内容赋值给 b。

二．填空题

1.以下程序运行后的输出结果是 \_\_\_\_。

```
#include<stdio.h>  
  
main()  
{ int a = 200 , b = 010 ;  
  
printf("%d%d\n" , a , b) ;  
  
}
```

参考答案：2008

【解析】整型变量 a 的值为 200，b 的值 "010" 是用八进制表示的 "10" 即十进制的 "8"，最后输出格式均为 %d，即十进制格式，所以输出为 "2008"。

2. 有以下程序 (说明：字符 0 的 ASCII 码值为 48)

```
#include<stdio.h>

main()
{
    char c1, c2;
    scanf("%d", &c1);
    c2=c1+9;
    printf("%c%c\n", c1, c2);
}
```

若程序运行时从键盘输入 48<回车>，则输出结果为\_\_\_\_\_。

参考答案：09

【解析】本题中通过 ASCII 码对字符变量赋值，由于字符 0 的 ASCII 码是 48，因此字符变量 c1 为 0，c2 的 ASCII 码比 c1 大 9，因此 c2 为 9。

#### 第04章 选择结构

##### 一. 选择题

1. 若变量已正确定义，在 if(W)printf("%d\n", k); 中，以下不可替代 W 的是。

A)a<>b + c    B)ch = getchar()    C) a = = b + c    D)a + +

【解析】在 C 语言中，表示不等于不能用 "<>"，而只能使用 "!="。其他选项均满足题目的要求。

2. 当变量 c 的值不为 2、4、6 时，值也为 "真" 的表达式是。

A)(c == 2)|| (c == 4)|| (c == 6)  
B)(c >= 2 && c <= 6) || (c != 3) || (c != 5)  
C)(c >= 2 && c <= 6) && ! (c % 2)  
D)(c >= 2 && c <= 6) && (c % 2 != 1)

【解析】满足表达式 (c >= 2 && c <= 6) 的整型变量 c 的值是 2,3,4,5,6。当变量 c 的值不为 2,4,6 时，其值只能为 3 或 5，所以表达式 c != 3 和 c != 5 中至少有一个为真，即不论 c 为何值，选项 B) 中表达式的值都为 "真"。正确答案为 B)。

3. 已知字母 A 的 ASCII 码值为 65，若变量 kk 为 char 型，以下不能正确判断出 kk 中的值为大写字母的表达式是。

A)kk >= A && kk <= Z  
B)! (kk >= A || kk <= Z)  
C)(kk + 32) >= a && (kk + 32) <= Z  
D)isalpha(kk) && (kk < 91)

【解析】C 语言的字符以其 ASCII 码的形式存在，所以要确定某个字符是大写字母，只要确定它的 ASCII 码在 'A' 和 'Z' 之间就可以了，选项 A) 和 C) 符合此要求。在选项 D) 中，函数 isalpha 用来确定一个字符是否为字母，大写字母的 ASCII 码值的范围为 65 到 90，所以如果一个字母的 ASCII 码小于 91，那么就能确定它是大写字母。本题答案选 B)。

4. 有以下程序：

```
#include <stdio.h>

main()
{
    int a = 1, b = 2, c = 3, d = 0;
    if(a == 1 && b++ == 2)
        if(b != 2 || c-- != 3)
            printf("%d, %d, %d\n", a, b, c);
    else printf("%d, %d, %d\n", a, b, c);
}
```

```
        else printf("%d  , %d , %d\n" , a , b , c) ;  
    }
```

程序运行后的输出结果是 \_\_\_\_\_。

A)1,2,3

B)1,3,2

C)1,3,3

D)3,2,1

【解析】 程序首先执行第一个判断语句， a = = 1成立；b + + = = 2，先判断 b是否为 2，再进行 b加1操作，执行后 b的值为 3。继续执行第二个判断语句，注意程序此时进行逻辑或运算， b != 2的值为真，因此条件表达式 b != 2||c - - != 3的值为真，程序便不再执行第二个逻辑语句 c - - != 3，而直接进行输出操作。因此，本题正确答案为 C)。

5.执行以下程序段后， w的值为 \_\_\_\_\_。

```
int w = 'A' , x = 14 , y = 15 ;  
w = ((x > y)&&(w < 'a')) ;
```

A) - 1

B)NULL

C)1

D)0

【解析】根据题目所给条件可知， x||y为1，w<'a'为1,1&&1 结果为1。因此正确答案为 C)。

6.若变量已正确定义，有以下程序段：

```
int a = 3 , b = 5 , c = 7 ;  
if(a > b) a = b ; c = a ;  
if(c != a) c = b ;  
printf("%d  , %d , %d\n" , a , b , c) ;
```

其输出结果是

A) 程序段有语法错

B)3,5,3

C)3,5,5

D)3,5,7

【解析】两个 if语句的判断条件都不满足，程序只执行了 c = a这条语句，所以变量 c的值等于 3，变量 b的值没能变化，程序输出的结果为 3,5,3。所以正确答案为选项 B)。

7.有以下程序段：

```
int a , b , c ;  
a = 10 ; b = 50 ; c = 30 ;  
if(a > b)a = b , b = c ; c = a ;  
printf("a = %d b = %d c = %d\n" , a , b , c) ;
```

程序的输出结果是 \_\_\_\_\_。

A)a = 10 b = 50 c = 10

B)a = 10 b = 50 c = 30

C)a = 10 b = 30 c = 10

D)a = 50 b = 30 c = 50

【解析】该题考查 if条件语句，当条件为真时，将 b值赋给 a，c值赋给 b；当条件为假时，将 a值赋给 c。该题中 if条件为假，所以将a的值赋给 c，故 a = 10 b = 50 c = 10。

8.有以下程序：

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

```

main()
{
    int x = 1, y = 2, z = 3;

    if(x > y)

    if(y < z) printf("%d", ++z);

    else printf("%d", ++y);

    printf("%d\n", x++);
}

```

程序的运行结果是 \_\_\_\_\_。

A)331

B)41

C)2

D)1

【解析】该题目考查 if 条件语句。else 语句和最近的一个 if 语句配对。由于 x > y 为假，所以直接执行最后一行代码。

9. 有以下程序：

```

#include <stdio.h>

main()
{
    int x = 1, y = 0, a = 0, b = 0;

    switch(x)
    {
        case 1 :
            switch(y)
            {
                case 0: a++; break;

                case 1: b++; break;

            }

            case 2 : a++; b++; break;

            case 3 : a++; b++;

        }

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

    }
}

```

程序的运行结果是

A)a = 1, b = 0

B)a = 2, b = 2

C)a = 1, b = 1

D)a = 2, b = 1

【解析】本题考查了 switch 结构的内容。在 C 语言中，程序执行完一个 case 标号的内容后，如果没有 break 语句，控制结构会转移到下一个 case 继续执行，因为 case 常量表达式只是起语句标号作用，并不是在该处进行条件判断。本题程序在执行完内部 switch 结构后，继续执行了外部 switch 结构的 case 2 分支。最后 a 和 b 的值分别为 2 和 1。

## 二．填空题

1. 在 C 语言中，当表达式值为 0 时表示逻辑值 "假"，当表达式值为 \_\_\_\_\_ 时表示逻辑值 "真"。

参考答案：非 0

【解析】本题一定要明确一个概念，即：所有非 0 的数（并不只是 "1"），在 c 语言中都表示逻辑值为 "真"。

2. 设 x 为 int 型变量，请写出一个关系表达式 \_\_，用以判断 x 同时为 3 和 7 的倍数时，关系表达式的值为真。

参考答案：(x%3==0)&&(x%7==0)

【解析】x%3==0 能保证 x 是 3 的倍数，x%7==0 能保证 x 是 7 的倍数，( x%3==0 ) && ( x%7==0 ) 能保证 x 是 3 的倍数并且

x 是 7 的倍数。

3. 有以下程序

```
#include<stdio.h>

main()
{   int a=1,b=2,c=3,d=0;
    if(a==1)
        if(b!=2)
            if(c==3)d=1;
            else    d=2;
            else if(c!=3)d=3;
            else d=4;
        else d=5;
    printf("%d\n",d);
}
```

程序运行后的输出结果是 \_\_\_\_。

参考答案： 4

【解析】 a==1 为真， b!=2 为假， c!=3 为假，所以 d=4。

4. 有以下程序

```
#include<stdio.h>

main()
{   int x ;
    scanf("%d" , &x) ;
    if(x>15)printf("%d" , x-5) ;
    if(x>10)printf("%d" , x) ;
    if(x>5)printf("%d\n" , x+5) ;
}
```

若程序运行时从键盘输入 12< 回车>，则输出结果为 \_\_\_\_\_。

参考答案： 1217

【解析】 本题中输入 12 时，第一条 if 的条件 x>15 不满足因此不执行，第二条 if 的条件 x>10 满足因此输出 12，第三条 if 的条件 x>5 满足因此输出 17。

5. 以下程序运行后的输出结果是 \_\_\_\_\_。

```
#include<stdio.h>

main()
{   int x=10 , y=20 , t=0 ;
    if(x==y)t=x ; x=y ; y=t ;
    printf("%d%d\n" , x , y) ;
}
```

参考答案： 20 0

【解析】 本题中 if 选择的条件 x==y 不满足，因此直接执行后面的 x=y 和 y=t，则 x 的值为 20，y 的值为 0。

## 第05章 循环结构

### 一．选择题

1. 以下不构成无限循环的语句或语句组是

A)n=0 ;

do{++n ; }while(n<=0) ;



```
B)n=0 ;
    while(1){n++ ; }

C)n=10 ;
    while(n) ; {n-- ; }

D)for(n=0 , i=1 ;; i++)
    n+=i ;
```

【解析】项 A) 中变量 n 的值，先自加 1，再进行循环条件判断，此时循环条件 n<= 0 不成立，跳出循环。所以正确答案为 A)。

2. 有以下程序

```
#include<stdio.h>

main()
{   int y = 10 ;
    while(y-- ) ;
    printf("y = %d\n" , y) ;
}
```

程序执行后的输出结果是

- A)y = 0            B)y = - 1  
C)y = 1            D)while 构成无限循环

【解析】当 y 减为 1 时判断 while (y-- )，此时满足条件，但是 y 变成 0。下次循环判断 while (y-- ) 时，因为 y 为 0 不满足条件跳出循环，但是此时也要执行 y--，所以 y 变成了 -1。打印输出时输出 -1。

3. 有以下程序：

```
#include <stdio.h>

main()
{   int i = 5 ;
    do
    {   if(i%3 == 1)
        if(i%5 == 2)
        {   printf("**%d" , i) ;   break ;   }
        i++ ;
    }   while(i != 0) ;
    printf("\n") ;
}
```

程序的运行结果是 \_\_\_\_\_。

- A)\*7  
B)\*3\*5  
C)\*5  
D)\*2\*6

【解析】整个程序中只有对 i 增加的语句而没有对 i 减少的语句，所以 2、3 都不可能出现，选项 B) 和 D) 错误。而 i= 5 时第一个 if 语句的表达式为假，所以选项 C) 也错误。

4. 以下程序段中的变量已正确定义：

```
for(i = 0 ; i<4 ; i++ , i++ )
    for(k = 1 ; k<3 ; k++ ) ; printf("**") ;
```

程序段的输出结果是 \_\_\_\_\_。

- A)\*\*\*\*\*  
B)\*\*\*\*

- C)\*\*  
D)\*

【解析】 本题考察 for 循环语句，注意第二个 for 语句的后面有一个分号，即 printf 函数不属于循环体，无论循环执行多少次，printf("") 语句只执行一次。因此，本题正确答案为 D)。

5. 有以下程序

```
#include<stdio.h>

main()
{ int a=1,b=2;
  for(;a<8;a++){b+=a;a+=2;}
  printf("%d,%d\n",a,b);
}
```

程序运行后的输出结果是

- A)9,18                  B)8,11                  C)7,11                  D)10,14

【解析】 初始值 a=1，b=2，第一次循环：b=b+a=2+1=3，a=a+2=1+2=3，a=a+1=3+1=4；第二次循环：b=b+a=3+4=7，a=a+2=4+2=6，a=a+1=6+1=7；第三次循环：b=b+a=7+7=14，a=a+2=7+2=9，a=a+1=9+1=10，故本题答案选 D。

6. 有以下程序：

```
#include <stdio.h>

main()
{ int i, j;
  for(i = 3 ; i>= 1 ; i - - )
  { for(j = 1 ; j<= 2 ; j + + )printf("%d" , i + j) ;
    printf("\n") ;
  }
}
```

程序的运行结果是 \_\_\_\_\_。

- A)2 3 4                  B)4 3 2  
3 4 5                  5 4 3  
C)2 3                  D)4 5  
3 4                  3 4  
4 5                  2 3

【解析】 该题目主要考查 for 嵌套循环，要注意循环变量 i 和 j 的取值范围。输出结果为变量 i 和 j 的和。

8. 设变量已正确定义，以下不能统计出一行中输入字符个数（不包含回车符）的程序段是 \_\_\_\_\_。

- A)n = 0 ; while((ch = getchar()) != '\n')n + + ;  
B)n = 0 ; while(getchar() != '\n')n + + ;  
C)for(n = 0 ; getchar() != 'n' ; n + + );  
D)n = 0 ; for(ch = getchar() ; ch != '\n' ; n + + ) ;

【解析】 A) 项每进行一次循环判断一个字符是否为回车，如果不是就进行下一次判断，因此可以统计出输入字符个数； B) 项与 A) 项的判断相似，仅为是否将读入的数据赋值给一个变量，因此并不影响判断结果； C) 想用 for 循环来判断，与 B) 项的判断完全相同，可以统计出输入字符个数； D) 项中 ch = getchar() 是给变量 ch 赋初值，如果输入回车，则程序只循环一次，如果输入一个非回车的字符，则程序进入死循环。因此，本题正确答案为 D)。

9. 有以下程序

```
#include<stdio.h>

main()
{ int i,j,m=1;
```

```

for(i=1;i<3;i++)
{ for(j=3;j>0;j--)
    {if(i*j>3)break;
      m*=i*j;
    }
  }
printf("m=%d\n",m);
}

```

程序运行后的输出结果是

A)m=6                  B)m=2                  C)m=4                  D)m=5

【解析】第一次循环 i=1，j=3和j=2时都能执行 m\*=i\*j，此时得到 m的值为6；然后进行第二次循环 i=2,j=3时会执行 break 语句，内部循环直接结束，此时 i再加1，也会导致退出外部循环，所以最终结果 m的值为6。

10.有以下程序

```

#include<stdio.h>

main()
{ int s ;
  scanf("%d" , &s) ;
  while(s>0)
  { switch(s)
    { case 1 : printf("%d" , s+5) ;
      case 2 : printf("%d" , s+4) ; break ;
      case 3 : printf("%d" , s+3) ;
      default : printf("%d" , s+1) ; break ;
    }
    scanf("%d" , &s) ;
  }
}

```

运行时，若输入 123450< 回车 >，则输出结果是

A)6566456                  B)66656                  C)66666                  D)6666656

【解析】switch 语句执行完一个 case 后面的语句后，流程控制转移到下一个 case 语句继续执行，遇到 break 会跳出本次循环。本题中输入 1时会输出 65，输入 2时会输出 6，输入 3时会输出 64，输入 4时会输出 5，输入 5时会输出 6，在输入 0时不满足循环条件，程序执行结束。

11.有以下程序段

```

int i , n;

for(i=0 ; i<8 ; i++)
{ n=rand()%5 ;
  switch(n)
  { case 1 :
    case 3 : printf("%d\n" , n) ; break ;
    case 2 :
    case 4 : printf("%d\n" , n) ; continue ;
    case 0 : exit(0) ;
  }
  printf("%d\n" , n) ;
}

```

```
}
```

以下关于程序段执行情况的叙述，正确的是

- A)for 循环语句固定执行 8次
- B) 当产生的随机数 n为4时结束循环操作
- C)当产生的随机数 n为 1和2时不做任何操作
- D)当产生的随机数 n为 0时结束程序运行

【解析】当产生的随机数 n为1或2时，由于没有遇到 break，因此程序将分别顺序执行 case 3 与 case 4；当 n为 3时，程序输出 n的值后跳出 switch 语句；当 n为4时，程序输出 n的值后将继续执行下一次循环；当 n为0时，程序执行 exit函数结束运行。

## 二．填空题

### 1.有以下程序

```
#include<stdio.h>

main()
{   int m,n;

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

    while(m!=n)
    {   while(m>n)m=m-n;

        while(m<n)n=n-m;

    }

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

}
```

程序运行后，当输入 14<空格>7<回车>时,输出结果是\_\_\_\_\_。

参考答案： 7

【解析】执行 while(m<n) 的循环，得到 m=14,n=7，然后执行 while(m>n) 的循环，得到 m=7,n=7。此时所有循环结束，m=7。

### 2.以下程序运行后的输出结果是\_\_\_\_\_。

```
#include <stdio.h>

main()
{   int a = 1 , b = 7 ;

    do{

        b = b/2; a + = b ;

    } while(b>1) ;

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

参考答案： 5

【解析】第一次循环执行后，b = 3，a = 4，满足条件 b>1，循环继续；第二次循环执行后，b = 1，a = 5，不满足条件 b>1，结束循环。所以输出的 a 值为 5。

### 3.有以下程序：

```
#include<stdio.h>

main()
{   int f , f1 , f2 , i ;

    f1 = 0 ; f2 = 1 ;

    printf("%d %d " , f1 , f2) ;

    for (i = 3 ; i<= 5 ; i + + )

    {   f = f1 + f2 ; printf("%d" , f) ;

        f1 = f2 ; f2 = f ;

    }

}
```

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

程序运行后的输出结果是 \_\_\_\_。

参考答案： 0 1 123

【解析】 第一个 printf 语句输出结果为 0 1。循环体总共循环三次，第一次循环结果为， f = 1，f1 = 1，f2 = 1；第二次循环结果为 f = 2，f1 = 1，f2 = 2；第三次循环结果为 f = 3，f1 = 2，f2 = 3。

4.若有定义： int k；，以下程序段的输出结果是 \_\_\_\_。

```
for(k = 2 ; k<6 ; k + + , k + + ) printf("##%d" , k) ;
```

参考答案： ##2##4

【解析】在 for 循环语句中，自变量 k 的自增表达式为 k + +，k + +。这是一个逗号表达式，所以输出结果为 ##2##4。

5.以下程序运行后的输出结果是 \_\_\_\_。

```
#include <stdio.h>  
  
main()  
{ int k = 1 , s = 0 ;  
  
do{  
    if((k%2) != 0)continue ;  
  
    s + = k; k + + ;  
  
}while(k>10) ;  
  
printf("s = %d\n" , s) ;  
  
}
```

参考答案： s=0

【解析】 continue 语句用于跳出本次循环，直接进行下一次循环。进行 if 判断时 k 的值为 1 不能够被 2 整除，因此执行 continue 退出本次循环，不对 s 进行任何操作，直接进行循环条件判断，此时 k 的值为 1 不满足循环条件，退出 while 循环，输出 s 的值为 0。

## 第06章 数组

### 一．选择题

1.下列选项中 ,能正确定义数组的语句是

A)int num[0..2008];                      B)int num[];  
C)int N=2008;                              D)#define N 2008  
int num[N];                                  int num[N];

【解析】选项 A 不符合 C 语言的语法要求；选项 B 中没有指定数组的大小；选项 C 中数组大小不能用变量来指定。

2.下列定义数组的语句中，正确的是。

A)int N = 10 ;    B)#define N 10    C)int x[0..10] ;    D)int x[] ;  
int x[N] ;              int x[N] ;

【解析】 A ) 中的 N 是一个变量，不可以用变量来定义数组，所以选项 A ) 错误。 C ) 中把所有的下标均列出不正确，此处只需指明数组长度即可。 D ) 中在定义时没有指明数组长度不正确，如果不指明长度就应在定义时对数组元素进行赋值，而此选项没有，所以错误。故本题答案为 B )。

3.若要定义一个具有 5 个元素的整型数组，以下错误的定义语句是

A)int a[5] = {0} ;              B)int b[] = {0,0,0,0,0} ;  
C)int c[2 + 3] ;                  D)int i = 5 , d[i] ;

【解析】在进行数组的定义时，不能使用变量对数组的长度进行定义。其他选项均符合数组定义标准。

4.以下错误的定义语句是。

A)int x[][3] = {{0} , {1} , {1,2,3}} ;  
B)int x[4][3] = {{1,2,3} , {1,2,3} , {1,2,3} , {1,2,3}} ;  
C)int x[4][] = {{1,2,3} , {1,2,3} , {1,2,3} , {1,2,3}} ;

D)int x[][3] = {1,2,3,4} ;

【解析】本题考查的是二维数组的定义和初始化方法。C语言中，在定义并初始化二维数组时，可以省略数组的第一维的长度，但是不能省略第二维的长度。故选项 C)错误。

5.若有定义：int a[2][3] ;, 以下选项中对 a数组元素正确引用的是。

A)a[2][ ! 1]

B)a[2][3]

C)a[0][3]

D)a[1>2][ ! 1]

【解析】C语言中数组下标是从 0开始的，所以二维数组 a[2][3] 的第一维下标取值为 0、1；第二维的下标取值为 0、1、2，因而选项 A)、B)、C)都是错误的，选项 D)表示的是数组元素 a[0][0]。所以正确答案为 D)。

6.有以下程序：

```
#include <stdio.h>

#include <string.h>

main()

{ char a[10] = "abcd" ;

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

}
```

程序运行后的输出结果是 \_\_\_\_\_。

A)7,4          B)4,10

C)8,8          D)10,10

【解析】strlen() 用来返回字符串的长度，而 sizeof() 返回的是一个对象或者类型所占的内存字节数，即数组所占的内存。

## 二．填空题

1.以下程序运行后的输出结果是 \_\_\_\_\_。

```
#include <stdio.h>

main()

{ int i,n [ 5 ] ={0} ;

for(i=1 ; i<=4 ; i++)

{ n [ i ] =n [ i-1 ] *2+1 ; printf("%d" , n [ i ] ) ; }

printf("\n") ;

} 参考答案： 13715
```

【解析】本题中已知 n [ 0 ] 为 0，根据循环的执行可以得到 n [ 1 ] 为 1，n [ 2 ] 为 3，n [ 3 ] 为 7，n [ 4 ] 为 15，因此输出结果是 13715。

2.有以下程序

```
#include<stdio.h>

main()

{ int i , n[] = {0,0,0,0,0} ;

for(i = 1 ; i<= 4 ; i + + )

{n[i] = n[i - 1]*3 + 1 ; printf("%d " , n[i]) ; }

}
```

程序运行后的输出结果是 \_\_\_\_\_。 参考答案： 1 4 13 40

【解析】第一次循环结果为：n [ 1 ] =0\*3+1, 即n [ 1 ] 的值为 1，打印输出 1；第二次循环结果为：n [ 2 ] =1\*3+1, 即n [ 2 ] 的值为 4，打印输出 4；第三次循环结果为：n [ 3 ] =4\*3+1, 即n [ 3 ] 的值为 13，打印输出 13；第四次循环结果为：n [ 4 ] =13\*3+1, 即n [ 4 ] 的值为 40，打印输出 40。

3.有以下程序

```

#include<stdio.h>

main()

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

    int b[3]={0},i;

    for(i=0;i<3;i++)      b[i]=a[i][2]+a[2][i];

    for(i=0;i<3;i++)      printf("%d",b[i]);

    printf("\n");

}

```

程序运行后的输出结果是 \_\_\_\_。 参考答案： 101418

【解析】当 i=0时， $b[0]=a[0][2]+a[2][0]=3+7=10$ ；当 i=1时， $b[1]=a[1][2]+a[2][1]=6+8=14$ ；当 i=2时， $b[2]=a[2][2]+a[2][2]=9+9=18$ ，则打印结果为 101418。

4.有以下程序

```

#include<stdio.h>

main()

{   int n[2] , i , j ;

    for(i = 0 ; i<2 ; i + + )n[i] = 0 ;

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

        for(j = 0 ; j<2 ; j + + )n[j] = n[i] + 1 ;

    printf("%d\n" , n[1]) ;

}

```

结果为：\_\_\_\_\_ 参考答案： 3

【解析】通过第一次  $\text{for}(i=0;i<2;i++) \text{ n}[i]=0$ ; 的循环语句可知，已经将数组  $\text{n}$  中的两个元素都赋初值为 0。接下来的循环中，第一次循环，外循环  $i=0$  的前提下：内循环  $j=0$  时，运行完  $\text{n}[0]=\text{n}[0]+1$  后  $\text{n}[0]$  为 1； $j=1$  时，运行完  $\text{n}[1]=\text{n}[0]+1$  后， $\text{n}[1]$  为 2。第二次循环，外循环  $i=1$  的前提下，内循环  $j=0$  时，运行完  $\text{n}[0]=\text{n}[1]+1$  后  $\text{n}[0]$  为 3； $j=1$  时，运行完  $\text{n}[1]=\text{n}[1]+1$  后， $\text{n}[1]$  为 3。

5.以下 fun函数的功能是在 N行M列的整型二维数组中，选出一个最大值作为函数值返回，请填写。（设M，N已定义）

```

int fun(int a[N][M])

{   int i , j , row = 0 , col = 0 ;

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

        for(j = 0 ; j<M ; j + + )

            if(a[i][j]>a[row][col]){row = i ; col = j ; }

    return( ____ ) ;

}

```

参考答案：  $\text{a}[\text{row}][\text{col}]$

【解析】通过程序可以看出，外循环是行，内循环是列。先在行不变的情况下找一行内最大的数据进行记录。通过语句  $\text{if}(\text{a}[\text{i}][\text{j}]>\text{a}[\text{row}][\text{col}])\{\text{row}=\text{i}; \text{col}=\text{j};\}$  可知，如果变量  $\text{a}[\text{i}][\text{j}]$  大于  $\text{a}[\text{row}][\text{col}]$ ，将  $\text{i}$  赋给了  $\text{row}$ ，将  $\text{j}$  赋给了  $\text{col}$ ，所以  $\text{a}[\text{row}][\text{col}]$  是记录当前最大值的变量。

6.有以下程序

```

#include<stdio.h>

main()

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

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

        for(j=i;j<3;j++)printf("%d",a[i][j]);

    printf("\n");

}

```

}

程序运行后的输出结果是 \_\_\_\_。 参考答案： 123569

【解析】当 i=0 时，j=0、1、2 时满足条件，则打印 a[0][0]、a[0][1]、a[0][2]；当 i=1 时，j=1、2 时满足条件，则打印 a[1][1]、a[1][2]；当 i=2 时，j=2 时满足条件，则打印 a[2][2]，所以打印结果为 123569。

7.有以下程序：

```
#include <stdio.h>

main()
{ char a[20] = "How are you ?" , b[20] ;
  scanf("%s" , b) ; printf("%s %s\n" , a , b) ;
}
```

程序运行时从键盘输入： How are you ? <回车>

则输出结果为 \_\_\_\_。 参考答案： How are you? How

【解析】用 %s 格式输入字符串时，空格和回车都作为输入数据的分隔符，不能被读入，因此数组 b 的输出结果为 How。

## 第07章 指针

### 一．选择题

1.有以下程序：

```
#include <stdio.h>

main()
{ int m = 1 , n = 2 , *p = &m , *q = &n , *r ;
  r = p ; p = q ; q = r ;
  printf("%d , %d , %d , %d\n" , m , n , *p , *q) ;
}
```

程序运行后的输出结果是 \_\_\_\_\_。

- A)1,2,1,2      B)1,2,2,1  
C)2,1,2,1      D)2,1,1,2

【解析】 m 和 n 的值不变，输出结果为 1,2。指针 \*p 和 \*q 交换了指向的位置，即 \*p = &n , \*q = &m，输出结果分别为 2,1。

2.以下程序段完全正确的是

- A)int \*p ;    scanf("%d" , &p) ;  
B)int \*p ;    scanf("%d" , p) ;  
C)int k , \*p = &k ;    scanf("%d" , p) ;  
D)int k , \*p ;    \*p = &k ;    scanf("%d" , p) ;

【解析】 A ) 选项输入的是指针型变量 p 的地址，变量一定义就已分配好了地址不能再指定了，所以 A 错误。 B ) 选项没有指定指针 p 应该指向的变量，没给变量赋初值。 D ) 选项中， p 是地址， \*p 是地址内存放的数据，它把整型变量 k 的地址赋给了 \*p，所以错误。

3.以下定义语句中正确的是。

- A)int a = b = 0;  
B)char A = 65 + 1 , b = b ;  
C)float a = 1 , \*b = &a , \*c = &b;  
D)double a = 0.0 ; b = 1.1 ;

【解析】 本题考查变量的定义方法。如果要一次进行多个变量的定义，则在它们之间要用逗号隔开。因此选项 A)和 D)错误；在选项 C)中，变量 c 是一个浮点型指针，它只能指向一个浮点型数据，不能指向指针变量 b；所以正确答案为 B)。

4.若有定义语句： double x , y , \*px , \*py ; 执行了 px = &x ; py = &y ; 之后，正确的输入语句是 \_\_\_\_\_。

- A)scanf("%f%f" , x , y);  
B)scanf("%f%f"&x , &y) ;



- C)scanf("%lf%le" , px , py);
- D)scanf("%lf%lf" , x , y) ;

【解析】 本题考查 scanf 函数，其格式为： scanf(" 控制格式 "，地址表列 )。其中地址表列中应是要赋值变量的地址。本题要为变量 x和y赋值，并定义了两个指针分别指向 x和y，因此取得变量 x和y的地址的方法有两种，一种是使用取地址符号 "&"，即 &x和&y，另一种是使用指针变量，即 px和py。选项A)和D)中地址表列表示错误； 选项 B)中控制格式与地址表列之间应用逗号分开。因此，本题正确答案为 C)。

5.有以下程序：

```
#include <stdio.h>

main()
{
    int a[] = {1,2,3,4} , y , *p = &a[3] ;
    --p ; y = *p ; printf("y = %d\n" , y) ;
}
```

程序的运行结果是

- A)y = 0
- B)y = 1
- C)y = 2
- D)y = 3

【解析】在程序中指针变量 p初始指向 a[3]，执行 p减1后，p指向a[2]，语句 y = \*p的作用是把 a[2]的值赋给变量 y，所以输出为y= 3。正确答案为 D)。

6.下列函数的功能是 \_\_\_\_\_。

```
fun(char *a , char *b)
{
    while((*b = *a) != '\0') {a ++ ; b ++ ; }
}
```

- A)将a所指字符串赋给 b所指空间
- B)使指针 b指向a所指字符串
- C)将a所指字符串和 b所指字符串进行比较
- D)检查 a和b所指字符串中是否有 '\0'

【解析】 表达式 \*b = \*a是将 a所指的字符赋给 b所指的空间，然后，指针 a和b依次后移，直到到达指针 a所指字符串的结尾。

7.设有定义： char \*c ；，以下选项中能够使字符型指针 c正确指向一个字符串的是 \_\_\_\_\_。

- A)char str[] = "string" ; c = str ;
- B)scanf("%s" , c) ;
- C)c = getchar() ;
- D)\*c = "string" ;

【解析】 选项B)、 D)中，没有为字符串分配空间，因此不能用 c指向字符串。选项 C)中getchar() 函数返回一个字符，不能将字符赋给字符指针 c。选项 A)中首先定义了字符数组 str，然后将 str的首地址赋给指针 c，这样就使指针 c指向了字符串 "string"。

8.有以下程序

```
#include<stdio.h>

main()
{
    char s[] = "rstuv" ;
    printf("%c\n" , *s + 2) ;
}
```

程序运行后的输出结果是

- A)tuv
- B)字符t的 ASCII 码值
- C)t
- D)出错

【解析】 \*s+2 相当于 (\*s)+2即先取出 s所指的数据然后对其加 2，s是字符串的首地址，所以 \*s即s [ 0 ] 就是字符 "r"，所以在

它的ASCII码上加上数字2就变成了字母"t"的ASCII码，所以以%c格式输出，即为选项C)。

9. 下列语句组中，正确的是

- A)char \*s;s="Olympic";
- B)char s[7];s="Olympic";
- C)char \*s;s={"Olympic"};
- D)char s[7];s={"Olympic"};

【解析】若s被定义为指针，选项A表示将指针s指向一个字符串，选项C表示为指针s赋一个字符串的值，显然选项A正确，选项C错误；若s被定义为数组，且要为该数组直接赋值，则必须在定义时进行，因此选项B和选项D都错误。

二．填空题

1. 以下程序的功能是：借助指针变量找出数组元素中的最大值及其元素的下标值。请填空。

```
#include<stdio.h>

main()
{int a[10],*p,*s;

  for(p=a;p-a<10;p++)scanf("%d",p);

  for(p=a,s=a;p-a<10;p++)    if(*p>*s) s=_____;

  printf("index=%d\n",s-a);

}  参考答案： p
```

【解析】s指向当前最大的元素，当 \*p>\*s 时，表示当前 p所指向的那个元素比 s指向的元素大，则 s应该记录当前最大的元素的地址。

2. 有以下程序

```
#include<stdio.h>

main()
{   int a[]={1,2,3,4,5,6},*k[3],i=0;

    while(i<3)
    { k[i]=&a[2*i];

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

      i++;

    }

}
```

程序运行后的输出结果是 \_\_\_\_。 参考答案： 135

【解析】当 i=0时，k [ 0 ] =&a [ 0 ] ；当 i=1时，k [ 1 ] =&a [ 2 ] ；当 i=2时，k [ 2 ] =&a [ 4 ] ，则打印结果为 135。

3. 以下程序的功能是： 借助指针变量找出数组元素中最大值所在的位置并输出该最大值。 请在输出语句中填写代表最大值的输出项。

```
#include<stdio.h>

main()
{   int a[10] , *p , *s ;

    for(p = a ; p - a<10 ; p + + )   scanf("%d" , p) ;

    for(p = a , s = a ; p - a<10 ; p + + )   if(*p>*s)      s = p ;

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

}
```

参考答案： \*s

【解析】因为题目中有 if ( \*p>\*s ) s=p ; 语句，可知如果 p所指的元素的值比 s所指的元素的值大，就把指针 p的地址赋予指针 s，即 s指向当前值最大的元素，所以最后应该输出的元素的值为 \*s。

第08章 函数

一．选择题

1. 以下叙述正确的是。

- A)C 语言程序是由过程和函数组成的
- B)C 语言函数可以嵌套调用 ,例如： fun(fun(x))
- C)C 语言函数不可以单独编译
- D)C 语言中除了 main 函数 ,其他函数不可作为单独文件形式存在

【解析】 C 语言中只有函数没有过程，故选项 A 错误； C 语言可以嵌套调用，故选项 B 正确； C 语言函数可以单独编译成 .dll 文件，故选项 C 错误； C 语言中除 main 函数以外，其他函数可作为单独文件形式存在，故选项 D 也错误。

2.有以下程序：

```
#include <stdio.h>

int fun(int a , int b)
{ if(b==0) return a ;
  else return(fun(--a , --b)) ;
}

main()
{ printf("%d\n" , fun(4,2)) ; }
```

程序的运行结果是 \_\_\_\_\_。

- A)1
- B)2
- C)3
- D)4

【解析】 该题目考查函数参数传递以及 if 条件语句。当变量 b 为 0 时，将 a 的值返回给主调函数，因此 a 进行两次自减 1 后，将其值返回并输出。

3.下面的函数调用语句中 func 函数的实参个数是 \_\_\_\_\_。

```
func(f2(v1 , v2) , (v3 , v4 , v5) , (v6 , max(v7 , v8))) ;
```

- A)3
- B)4
- C)5
- D)8

【解析】 C 语言函数定义中，参数列表之间使用逗号分隔。该题目中 func 的函数列表中使用两个逗号，将 3 个参数分隔开。

4.有以下程序：

```
#include<stdio.h>

void fun(int *a , int *b)
{ int *c ;
  c = a ; a = b ; b = c ;
}

main()
{ int x = 3, y = 5, *p = &x, *q = &y ;
  fun(p , q) ; printf("%d , %d , " , *p , *q) ;
  fun(&x , &y) ; printf("%d , %d\n" , *p , *q) ;
}
```

程序运行后的输出结果是 \_\_\_\_\_。

- A)3,5,5,3
- B)3,5,3,5
- C)5,3,3,5
- D)5,3,5,3

【解析】子函数 fun 功能为交换形式参数的值，即交换指针变量 a 和 b 的值。但是 fun 函数并不能够交换实参的值，因此 fun(p, q) 不能交换 p 和 q 的值，所以第一个 printf 语句的输出为 3,5。第二个 fun 函数对 x 和 y 的地址进行了操作，同样不能交换 x 和 y 的值，并且不能影响 p 和 q 指针指向的数据，因此第二个 printf 语句的输出为 3,5。

5. 有以下程序：

```
#include<stdio.h>

int f(int x , int y)
{return((y - x)*x) ; }

main()
{ int a = 3 , b = 4 , c = 5 , d ;
  d = f(f(a , b) , f(a , c)) ;
  printf("%d\n" , d) ;
}
```

程序运行后的输出结果是 \_\_\_\_\_。

- A)10
- B)9
- C)8
- D)7

【解析】将 a、b、c 的值分别带入 f(a, b) 和 f(a, c) 中计算得：f(a, b) = 3, f(a, c) = 6，再计算 f(3,6)，算出 d = 9。

6. 有以下程序：

```
#include<stdio.h>

int fun(int x , int y)
{ if(x == y) return(x) ;
  else return((x + y)/2) ;
}

main()
{ int a = 4 , b = 5 , c = 6 ;
  printf("%d\n" , fun(2*a, fun(b,c)))
}
```

则运行结果为。

- A)3      B)6
- C)8      D)12

【解析】函数 fun 的功能是求两个整数的平均值，返回值仍为整数。5 和 6 的平均取整后为 5, 8 和 5 的平均取整后为 6。因此，本题答案为 B)。

7. 有以下程序

```
#include<stdio.h>

int f(int X) ;

main()
{ int n = 1 , m ;
  m = f(f(f(n))) ; printf("%d\n" , m) ;
}

int f(int x)
{ return x*2 ; }
```

程序运行后的输出结果是

- A)1      B)2      C)4      D)8

【解析】第一次调用的是最内层的  $f(n)$ ，即  $f(1)$  返回值 2。第二次调用中间的  $f(f(n))$ ，即  $f(2)$  返回值 4。最后调用最外层的  $f(f(f(n)))$ ，即  $f(4)$  返回值 8，最后打印输出。

## 二．填空题

1. 以下程序的输出结果是 \_\_\_\_\_。

```
#include <stdio.h>

void swap(int *a, int *b)
{
    int *t;

    t = a; a = b; b = t;
}

main()
{
    int i = 3, j = 5, *p = &i, *q = &j;

    swap(p, q); printf("%d %d\n", *p, *q);
}
```

参考答案：3 5

【解析】函数  $\text{swap}(\text{int } *a, \text{int } *b)$  的功能是实现  $*a$  和  $*b$  中两个数据的交换，在主函数中调用  $\text{swap}(p, q)$  后，参形指针变量  $a$  和  $b$  分别指向  $i$  和  $j$ ，在  $\text{swap}(\text{int } *a, \text{int } *b)$  执行完后，指针变量  $a$  和  $b$  分别指向  $j$  和  $i$ ，而指针变量  $p, q$  所指向变量的值没有发生变化，所以输出结果为 3 5。

2. 以下 fun 函数的功能是：找出具有 N 个元素的一维数组中的最小值，并作为函数值返回。请填写。 (设 N 已定义)

```
int fun(int x[N])
{
    int i, k = 0;

    for(i = 0; i < N; i++)
        if(x[i] < x[k]) k = _____;

    return x[k];
}
```

参考答案：i

【解析】通过函数中语句  $\text{return } x[k];$  可知  $x[k]$  表示的是该一维数组中最小的值，所以当  $x[i]$  小于  $x[k]$  所代表的数值后应将  $i$  变量赋给  $k$  变量，使得  $x[k]$  表示当前值较小的那个数据。

3. 有以下函数

```
void prt(char ch, int n)
{
    int i;

    for(i = 1; i <= n; i++)
        printf(i % 6 != 0 ? "%c":"%c\n", ch);
}
```

执行调用语句  $\text{prt}('*', 24);$  后，函数共输出了 \_\_\_\_\_ 行 \* 号。

参考答案：4

【解析】本题中输出 \* 并换行的条件是  $i$  能被 6 整除，否则输出 \* 不换行。因此只有在  $i$  为 6、12、18 和 24 时 \* 的输出才会换行，最终输出 4 行 \*。

4. 请将以下程序中的函数声明语句补充完整。

```
#include <stdio.h>

int _____;

main()
{
    int x, y, (*p)();

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

    p = max;
```

```

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

}

int max(int a , int b)

{ return(a>b ? a : b) ; }

```

参考答案： max(int a,int b)

【解析】本题考的是函数声明。函数声明是把函数的名字、函数类型以及形参的类型、个数和顺序通知给编译系统。函数在被调用前必须先声明，因为 main 函数后面有 max 函数的定义。

5. 有以下程序，请在 \_\_\_ 处填写正确语句，使程序可正常编译运行。

```

#include<stdio.h>

_____ ;

main()

{ double x , y , (*p)() ;

scanf("%lf%lf" , &x , &y) ;

p=avg ;

printf("%f\n" , (*p)(x , y)) ;

}

double avg(double a , double b)

{ return((a+b)/2) ; }

```

参考答案： double avg(double a ,double b);

【解析】c 语言编译器对于源代码的编译是顺序编译的，当任意一个函数中调用某个函数的时候，这个函数的原型必须在调用之前声明，否则就会出现编译错误。本题中需要在 main 函数调用 avg 之前对其进行声明。

6. 有以下程序：

```

#include <stdio.h>

fun(int x)

{ if(x/2>0) fun(x/2) ;

printf("%d ." , x) ;

}

main()

{ fun(6) , printf("\n") ; }

```

程序运行后的输出结果是 \_\_\_\_\_。

参考答案： 1 3 6

【解析】依次执行 fun(6) ， fun(3) ， fun(1) ，当执行 fun(6) 时要调用 fun(3) ，执行 fun(3) 时要调用 fun(1) ，所以输出的结果为 1 3 6。

## 第09章 变量的属性和编译预处理

### 一．选择题

1. 有以下程序：

```

#include<stdio.h>

int b = 2;

int fun(int *k)

{ b = *k + b ; return(b) ; }

main()

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

for(i = 2 ; i<4 ; i+ + ){b = fun(&a[i]) + b; printf("%d" , b) ; }

printf("\n") ;

```

}

程序运行后的输出结果是 \_\_\_\_\_。

- A)10 12            B)8 10  
C)10 28            D)10 16

【解析】b为全局变量，在第一次执行 for循环后 b的值变为 10并输出，第二次执行 for循环后 b的值变为 28。因此，本题答案选 C)。

2.在C语言中，只有在使用时才占用内存单元的变量，其存储类型是。

- A)auto 和register  
B)extern 和register  
C)auto 和static  
D)static 和register

【解析】在 C语言中只有自动变量和寄存器变量在使用时才占用内存单元。所以正确答案为 A)。

3.设函数中有整型变量 n，为保证其在未赋初值的情况下初值为 0，应选择的存储类别是 \_\_\_\_C\_\_\_\_\_。

- A)auto            B)register  
C)static            D)auto 或register

【解析】static(静态局部变量)：有时我们希望函数中局部变量的值在函数调用结束后不消失而保留原值，这时就将局部变量指定为静态局部变量，如果定义静态局部变量时不赋值，则编译时自动赋初值 0(对数据类型变量)或空字符(对字符变量)。

auto(自动变量)：函数中的局部变量，如不专门声明为 static存储类别，都是动态地分配存储空间的，数据存储在动态存储区中。函数中的形参和在函数中定义的变量都属此类，在调用该函数时系统会给它们分配存储空间，在函数调用结束时自动释放这些存储空间。

register(寄存器变量)：为了提高效率，C语言允许将局部变量的值放在 CPU中的寄存器中，这种变量叫寄存器变量。

4.有以下程序：

```
#include <stdio.h>

int f(int n) ;

main()
{ int a = 3 , s ;
  s = f(a) ; s = s + f(a) ; printf("%d\n" , s) ;
}

int f(int n)
{ static int a = 1 ;
  n + = a + + ;
  return n ;
}
```

程序运行后的输出结果是 \_\_\_\_C\_\_\_\_\_。

- A)7            B)8  
C)9            D)10

【解析】s = f(a)表达式第一次调用 f(n)函数时，f(n)函数运行结果为 n = 4，a = 2，即s等于 4；s = s + f(a)表达式第二次调用 f(n)函数时，f(n)函数运行结果为 n = 5，所以得到 s值为 9。

5.有以下程序

```
#include<stdio.h>

int fun()

{ static int x=1;
  x*=2; return x;
}
```

```

main()
{
    int i,s=1;

    for(i=1;i<=2;i++)    s=fun();

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

}

```

程序运行后的输出结果是

- A)0                      B)1                      C)4                      D)8

【解析】 x 被声明为静态变量，第一次调用 fun() 后，x 值为 2，第二次调用时直接执行 x\*=2, 值为 4。

6. 以下叙述中错误的是

- A) 在程序中凡是以 “#” 开始的语句行都是预处理命令行    D
- B) 预处理命令行的最后不能以分号表示结束
- C) #define    MAX 是合法的宏定义命令行
- D) C 程序对预处理命令行的处理是在程序执行的过程中进行的

【解析】 C 语言中的预处理命令以符号 # 开头，这些命令是在程序编译之前进行处理的，选项 D) 的描述错误。故答案选 D)。

7. 以下关于宏的叙述中正确的是      C。

- A) 宏名必须用大写字母表示
- B) 宏定义必须位于源程序中所有语句之前
- C) 宏替换没有数据类型限制
- D) 宏调用比函数调用耗费时间

【解析】 宏名可以大写，也可以小写。其有效范围为定义命令之后到本源文件结束。宏的调用是字符的替换，没有数据类型限制。

## 二．填空题

1. 以下程序的输出结果是      。

```

#include    <stdio.h>

int fun(int x)
{
    static int t = 0 ;

    return(t + = x) ;

}

main()
{
    int s , i ;

    for(i = 1 ; i< = 5 ; i + + ) s = fun(i) ;

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

}

```

参考答案： 15

【解析】 static 关键字用来声明 “静态局部变量”，静态局部变量属于静态存储类别，在静态存储区内分配存储单元。在程序运行期间不释放变量的值。

2. 以下程序运行后的输出结果是          。

```

#include<stdio.h>

#include    <stdlib.h>

#include<string.h>

main()
{
    char *p ;    int i ;

    p=(char*)malloc(sizeof(char)*20)    ;

```



```

        strcpy(p , "welcome") ;

        for(i=6 ; i>=0 ; i--)putchar(*(p+i)) ;

        printf("\n") ;    free(p) ;

    }

```

参考答案： emoclew

【解析】本题中用 malloc 函数申请一个 20 个字符大小的空间，并用指针 p 指向空间的首地址。 strcpy 函数使 "welcome" 字符串存放到分配的空间，最后用 putchar 函数倒序输出字符串的前 7 个字符。 p [ 0 ] = w , p [ 1 ] = e , p [ 2 ] = l , p [ 3 ] = c , p [ 4 ] = o , p [ 5 ] = m , p [ 6 ] = e。

## 第10章 复杂数据类型

### 一 . 选择题

1.有以下程序：

```

#include    <stdio.h>

main()

{ struct STU{   char   name[9] ;   char   sex ;   double   score[2] ;   } ;

    struct STU   a = {"Zhao" , 'm',85.0,90.0} , b = {"Qian" , 'f',95.0,92.0} ;

    b = a ;

    printf("%s , %c , %2.0f , %2.0f\n" , b.name , b.sex , b.score[0] , b.score[1]) ;

}

```

程序的运行结果是 \_\_\_\_D\_\_\_\_。

A)Qian , f,95,92

B)Qian , m,85,90

C)Zhao , f,95,92

D)Zhao , m,85,90

【解析】该题目考查结构体数据类型，注意语句 b = a，因此输出 b.name 为 zhao，b.sex 为 m，依此类推。

2.以下结构体类型说明和变量定义中正确的是。 A

A)typedef struct

{int n ; char c ; }REC ;

REC t1 , t2 ;

B)struct REC ;

{int n ; char c ; } ;

REC t1 , t2 ;

C)typedef struct REC ;

{int n=0 ; char c='A' ; }t1 , t2 ;

D)struct

{int n ; char c ; }REC ;

REC t1 , t2 ;

【解析】本题考查的是 typedef 的用法和结构体变量的定义方法。 typedef 可用于声明结构体类型，其格式为 "typedef struct { 结构元素定义 }结构类型；"。本题正确答案为 A)。

3.若有以下语句

```

typedef struct S

{ int g ; char h ; }T ;

```

以下叙述中正确的是

A)可用 S定义结构体变量

B)可用 T定义结构体变量

C)S 是 struct 类型的变量

D)T 是 struct S 类型的变量

【解析】此题考察的是结构体的定义方式。 S是我们定义的结构体的名字，在题目中顺便将 T定义为 struct S 类型，即 T被定义为一个类型名。这样就可以用 T来定义说明新的变量了。在此 S与T 都不是变量的名称。

4.下面结构体的定义语句中，错误的是 \_\_B\_\_\_\_\_。

- A)struct ord {int x ; int y ; int z ; } ; struct ord a ;
- B)struct ord {int x ; int y ; int z ; }struct ord a ;
- C)struct ord {int x ; int y ; int z ; }a ;
- D)struct {int x ; int y ; int z ; }a ;

【解析】 定义结构体变量有三种方式： 先声明结构体类型，再定义变量名，如选项 A)所示； 在声明类型的同时定义变量，如选项 C)所示； 直接定义结构体类型变量，如选项 D)所示。

5.有以下程序：

```
#include <stdio.h>

struct st
{
    int x , y ;
}data[2] = {1,10,2,20} ;

main()
{
    struct st *p = data ;
    printf("%d ", p ->y) ;    printf("%d\n", ( ++ p ) ->x) ;
}
```

程序的运行结果是 \_\_\_\_\_。

- A)10,1
- B)20,1
- C)10,2
- D)20,2

【解析】数组名的值即为数组首地址，所以 p ->y 可得第一个元素的 y值，( ++ p ) ->x 可得第二个元素的 x值。

6.设有定义：

```
struct complex
{
    int real,unreal;
}data1={1,8},data2;
```

则以下赋值语句中错误的是

- A)data2=data1;
- B)data2=(2,6);
- C)data2.real=data1.real;
- D)data2.real=data1.unreal;

【解析】选项 B 需要强制转换数据类型，应该为 (struct complex){2,6}; 。

7.有以下程序

```
#include<stdio.h>

struct ord
{
    int x , y ;
}dt [ 2 ]={1 , 2 , 3 , 4} ;

main()
{
    struct ord*p=dt ;
    printf("%d,",++(p->x)) ; printf("%d\n",++(p->y)) ;
}
```

程序运行后的输出结果是

- A)1,2
- B)4,l
- C)3,4
- D)2,3

【解析】本题中定义了一个结构体数组 dt [ 2 ]，其中 dt [ 0 ] .x=1，dt [ 0 ] .y=2，dt [ 1 ] .x=3，dt [ 1 ] .y=4。在main 函数中指针 p指向了结构体数组的第一个元素，因此 p->x 值为 1，p->y 值为 2，自加运算的结果分别为 2和3。

8.有以下程序

```
#include<stdio.h>

struct S

{ int a, b; }data [ 2 ] ={10,100,20,200};

main()

{ struct S p=data [ 1 ];

    printf("%d\n" , ++(p.a)) ;

}
```

程序运行后的输出结果是

- A)10                    B)11                    C)20                    D)21

【解析】本题中定义了一个结构体数组 data [ 2 ] ,其中 data [ 0 ].a=10 ,data [ 0 ].b=100 ,data [ 1 ].a=20 ,data [ 1 ].b=200 。在main 函数中结构体数组的第 2 个元素 data [ 1 ] 赋值给 p ,即p指向结构体数组的第 2 个元素 ,因此 p.a 的值为 20 ,进行自加运算后结果为 21。

9. 有以下定义和语句

```
struct workers

{int num;char name[20];char c;

    struct

    {int day;int month;int year;}s;

};

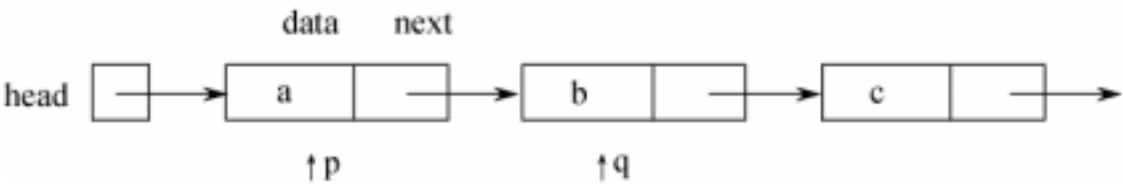
struct workers w,*pw;

pw=&w;
```

能给 w 中year 成员赋 1980 的语句是

- A)\*pw.year=1980;                    B)w.year=1980;  
C)pw->year=1980;                    D)w.s.year=1980;

【解析】结构体 structure workers 中的成员 s 是结构体类型 ,给 w 中成员 year 赋值的语句是 w.s.year=1980 ,故选 D。  
10.假定已建立以下链表结构 ,且指针 p 和 q 已指向如下图所示的结点 :



则以下选项中可将 q 所指结点从链表中删除并释放该结点的语句组是 \_\_\_\_\_。

- A)(\*p) . next=(\*q) . next ; free(p) ;  
B)p=q->next ; free(q) ;  
C)p=q ; free(q) ;  
D)p->next=q->next ; free(q) ;

【解析】要将结点 b 从链表中删除 ,应先将 a 的指针域指向 b 结点的下一个结点 ,即 p ->next = q ->next ,然后释放指针 q 的空间。

二 . 填空题

1. 有以下程序 :

```
#include <stdio.h>

typedef struct

{ int num ; double s ; }REC ;

void fun1(REC x) {x.num = 23 ; x.s = 88.5 ; }

main()

{ REC a = {16,90.0} ;

    fun1(a) ;
```

```
printf("%d\n" , a.num) ;  
}
```

程序运行后的输出结果是 \_\_\_\_\_。

参考答案： 16

【解析】 主函数中，通过 funl( ) 函数将 a 值传递给 x，但没有把形参 x 的值返回，此时变量 a 的值并没有发生变化，所以输出 a.num 的值为 16。

2. 设有定义：

```
struct person  
{ int ID ; char name[12] ; }p ;
```

请将 scanf("%d" , \_\_\_\_\_)；语句补充完整，使其能够为结构体变量 p 的成员 ID 正确读入数据。

参考答案： &p.ID

【解析】结构体成员的引用通过符号 "." 来表示，通过 scanf 语句对变量进行赋值时，要用取地址符 &。

3. 下列程序的运行结果为 \_\_\_\_\_。

```
#include <stdio.h>  
  
#include<string.h>  
  
struct A  
{ int a; char b[10]; double c; } ;  
  
void f(struct A *t) ;  
  
main()  
{ struct A a = {1001 , "ZhangDa" , 1098.0} ;  
f(&a) ; printf("%d , %s , %6.1f\n" , a.a , a.b , a.c) ;  
}  
  
void f(struct A *t)  
{ strcpy(t ->b , "ChangRong") ; }
```

参考答案： 1001,ChangRong,1098.0

【解析】此题考的是结构体用法。函数 f 的功能为将结构体的第二个变量修改为 ChangRong。主函数为运行 f 函数后，将结构体输出。

4. 以下程序把三个 NODETYPE 型的变量链接成一个简单的链表，并在 while 循环中输出链表结点数据域中的数据。请填空。

```
#include <stdio.h>  
  
struct node  
{ int data ; struct node *next ; } ;  
  
typedef struct node NODETYPE ;  
  
main()  
{ NODETYPE a , b , c , *h , *P ;  
a . data = 10 ; b.data = 20 ; c.data = 30 ; h = &a ;  
a . next = &b ; b.next = &c ; c.next = \0 ;  
p = h ;  
while(p){printf("%d , " , p ->data) ; __p=p->next  
____ ; }  
printf("\n") ;  
}
```

【解析】打印完一个链表结点的数据域中的数据域后，用 p = p ->next 使链表指针指向下一个链表结点。

第 11 章 文件

一．选择题

1. 下列关于 C 语言文件的叙述中正确的是 \_\_\_\_\_。

- A) 文件由一系列数据依次排列组成，只能构成二进制文件
- B) 文件由结构序列组成，可以构成二进制文件或文本文件
- C) 文件由数据序列组成，可以构成二进制文件或文本文件
- D) 文件由字符序列组成，其类型只能是文本文件

【解析】 文件由数据序列组成，可以构成二进制文件，也可以构成文本文件。

2. 有以下程序：

```
#include <stdio.h>

main()
{ FILE *fp ; int a[10] = {1,2,3} , i , n ;

  fp = fopen("d1.dat" , "w") ;

  for(i = 0 ; i<3 ; i+ + ) fprintf(fp , "%d" , a[i]) ;

  fprintf(fp , "\n") ;

  fclose(fp) ;

  fp = fopen("d1.dat" , "r") ;

  fscanf(fp , "%d" , &n) ;

  fclose(fp) ;

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

}
```

程序的运行结果是

- A)12300
- B)123
- C)1
- D)321

【解析】 在函数中首先把整型数组 a[10] 中的每个元素写入文件 d1.dat 中，然后再次打开这个文件，把文件 d1.dat 中的内容读入到整型变量 n 中，最后输出变量 n 的值。所以正确答案为 B)。

3. 有以下程序：

```
#include <stdio.h>

main()
{ FILE *pf ;

  char *s1 = "China" , *s2 = "Beijing" ;

  pf = fopen("abc.dat" , "wb + ") ;

  fwrite(s2,7,1 , pf) ;

  rewind(pf) ; /*文件位置指针回到文件开头 */

  fwrite(s1,5,1 , pf) ;

  fclose(pf) ;

}
```

以上程序执行后 abc.dat 文件的内容是 \_B\_\_\_\_\_。

- A)China
- B)Chinang
- C)ChinaBeijing
- D)BeijingChina

【解析】 该题目考查文件相关知识。 "wb + "用于打开或建立二进制文件并允许对其进行读和写操作。文件操作先写入了 s2，然后将文件指针移动到文件夹，再写入 s1，这样 s1 就会覆盖掉一部分 s2 的内容。

4. 有以下程序：

```
#include<stdio.h>

main()
{ FILE    *f ;

  f = fopen("filea.txt" , "w") ;

  fprintf(f , "abc") ;

  fclose(f) ;

}
```

若文本文件 filea.txt 中原有内容为： hello ，则运行以上程序后，文件 filea.txt 中的内容为 \_\_\_\_C\_\_\_\_。

- A)helloabc                      B)abclo  
C)abc                              D)abchello

【解析】 "w"表示建立一个供写入的文件。如果文件不存在，系统将用在 fopen 调用中指定的文件名建立一个新文件，如果指定的文件已存在，则将从文件的起始位置开始写入，文件中原有的内容将全部消失。

5. 有以下程序

```
#include<stdio.h>

main()
{ FILE *fp;char str[10];

  fp=fopen("myfile.dat","w");

  fputs("abc",fp);fclose(fp);

  fp=fopen("myfile.dat","a+");

  fprintf(fp,"%d",28);

  rewind(fp);

  fscanf(fp,"%s",str);puts(str);

  fclose(fp);

}
```

程序运行后的输出结果是 C

- A)abc                              B)28c  
C)abc28                              D)因类型不一致而出错

【解析】首先打开文件写入字符串 "abc"，然后关闭文件，再打开时文件指针定位到了最后，写入 "28"，然后重定位位置指针到开始，读取字符串为 "abc28"。

6. 设fp已定义，执行语句 fp=fopen("file","w") ；后，以下针对文本文件 file操作叙述的选项中正确的是。 B

- A) 写操作结束后可以从头开始读      B)只能写不能读  
C) 可以在原有内容后追加写              D)可以随意读和写

【解析】本题中用 "w" 方式打开文件，只能向文件写数据。如果原来不存在该文件，则新创建一个以指定名字命名的文件；如果已存在该文件，则把原文件删除后重新建立一个新文件，而不是把内容追加到原文件后。

## 二．填空题

1. 以下程序打开新文件 f.txt，并调用字符输出函数将 a数组中的字符写入其中，请填空。

```
#include<stdio.h>

main()
{ FILE _____ *fp ;

  char a[5] = {'1' , '2' , '3' , '4' , '5'} , i ;

  fp = fopen("f.txt" , "w") ;

  for(i = 0 ; i<5 ; i + + )fputc(a[i] , fp) ;

  fclose(fp) ;

}
```

```
}
```

【解析】在这里需要定义文件指针，定义文件指针的格式为：FILE \* 变量名。

2. 以下程序的功能是从名为 filea.dat 的文本文件中逐个读入字符并显示在屏幕上。请填空。

```
#include <stdio.h>

main()
{ FILE *fp ; char ch ;

  fp = fopen(____ "filea.dat","r" ____);

  ch = fgetc(fp) ;

  while( ! feof(fp)) { putchar(ch) ; ch = fgetc(fp) ; }

  putchar( \n ) ; fclose(fp) ;

}
```

【解析】fopen 函数的调用方式通常为 fopen( 文件名，使用文件方式 )。本题中要求程序可以打开 filea.dat 文件，并且读取文件中的内容。所以空白处应当填入 "filea.dat" ， "r"。

3. 以下程序用来判断指定文件是否能正常打开，请填空。

```
#include <stdio.h>

main()
{ FILE *fp ;

  if(((fp = fopen("test.txt" , "r")) == __NULL ____))

    printf(" 未能打开文件！ \n") ;

  else

    printf(" 文件打开成功！ \n") ;

}
```

【解析】本题考查 fopen 函数的用法。若 fopen 不能实现打开任务时，函数会带回一个出错信息，出错原因可能是磁盘出现故障，磁盘无法建立新文件等，此时 fopen 函数将带回一个空指针 NULL。因此通过判断返回值是否为 NULL 即可判断是否读取文件正确。

4. 以下程序运行后的输出结果是 \_\_\_\_123456 \_\_\_\_。

```
#include<stdio.h>

main()
{ FILE *fp ; int x [ 6 ] ={1 , 2 , 3 , 4 , 5 , 6} , i ;

  fp=fopen("test.dat" , "wb") ;

  fwrite(x , sizeof(int) , 3 , fp) ;

  rewind(fp) ;

  fread(x , sizeof(int) , 3 , fp) ;

  for(i=0 ; i<6 ; i++)printf("%d" , x [ i ] ) ;

  printf("\n") ;

  fclose(fp) ;

}
```

【解析】本题中 fwrite 函数向目标文件指针 fp 指向的文件 test.dat 中写入 3 个 int 数据，即 123。rewind 函数将文件内部的位置指针重新指向文件的开头。fread 函数将从 fp 所指文件中读取 3 个 int 数据到 x 指向的地址，因此数组 x 的元素没有变化。