

## C 语言程序设计模拟试题（六）

一、单项选择题（在每小题的四个备选答案中，选出一个正确的答案，并将其号码填在题干后的括号内。每小题 1 分，共 16 分）

1. 以下叙述中正确的是 C。  
A) C 程序中的注释只能出现在程序的开始位置  
B) C 程序要求一行内只能写一条语句  
C) C 程序一条语句可以写在多行上  
D) C 程序可从任何指定的函数开始运行
2. 计算机能直接执行的程序是 D。  
A) 源程序      B) 目标程序      C) 汇编源程序      D) 可执行程序
3. 按照 C 语言规定的用户标识符命名规则，不能出现在标识符中的是 B。  
A) 字母字符      B) 加号字符 '+'      C) 数字字符      D) 下划线字符
4. 在 C 语言中，合法的长整型常数是 A。  
A) 8L      B) 16u      C) 18\$      D) 168d
5. 表达式  $3.6 - 5 / 2 + 1.2 + 5 \% 2$  的值是 B。  
A) 4.3      B) 4.8      C) 3.3      D) 3.8  
*- 计算*
6. 以下关于 int 类型的变量占用内存大小的叙述中正确的是 D。  
A) 一定占用 4 个字节      B) 一定占用 2 个字节  
C) 由用户自己定义      D) 由 C 语言编译系统决定
7. 以下不合法的字符常量是 A。  
A) '\018'      B) '\n'      C) '\\'      D) '\xcc'  
*+ 十六进制*
8. 以下不合法的用户标识符是 C。  
A) j2\_      B) If      C) 4x      D) \_8
9. 设变量 x 和 y 均已正确定义并赋值，以下的 if 语句中，在编译时将产生错误信息的是 C。  
A) if (x++);  
B) if (x > y && y != 0);  
C) if (x > y) x--  
else y++;  
D) if (y < 0){;}  
else x++;
10. 在下列选项中，不正确的赋值语句是 D。  
A) ++x;      B) n1 = n2 = n3 = 0;      C) k = i = j;      D) b + c = 1;
11. 以下叙述中错误的是 B。  
A) C 语言是一种结构化程序设计语言  
B) 使用三种基本结构构成的程序只能解决简单问题  
C) 结构化程序由顺序、分支、循环三种基本结构组成  
D) 结构化程序设计提倡模块化的设计方法
12. 以下叙述中错误的是 C。

A) 对于 int 等类型的非字符数组, 不能直接用数组名对数组进行整体输入或输出

B) 数组名代表的是数组所占存储区的首地址, 实际是一个指针常量, 其值不可改变

C) 在程序执行中, 当数组元素的下标超出所定义的下标范围时, 系统一定会给出“下标越界”的出错信息

D) 可以通过初始化赋值的方式确定数组元素的个数

13. 设有如下说明:

```
typedef struct Complex  
{
```

```
    float real;
```

```
    float image;
```

```
} ComplexType;
```

则下面叙述中正确的是 C。

A) 以上的定义形式非法

B) Complex 是一个结构变量

C) ComplexType 是一个结构类型

D) ComplexType 是一个结构变量

14. 若程序有宏定义“#define N 8”, 则以下叙述中正确的是 B。

A) 宏定义中定义了整型标识符 N 的值为整数 8

B) 在编译程序对 C 源程序进行预处理时用 8 替换标识符 N

C) 对 C 源程序进行编译时用 8 替换标识符 N

D) 在运行时用 8 替换标识符 N

15. 读取二进制文件的函数调用形式为“fread(buffer, size, count, fp);”, 其中 fp 是 A。

A) 一个文件指针, 指向待读取的文件 fp

B) 一个整型变量, 代表待读取的数据的字节数

C) 一个内存块的首地址, 代表读入数据存放的地址 buffer

D) 一个内存块的大小

16. 若函数形参为整型, 函数调用时的实参为整型变量, 以下关于函数形参和实参的叙述中正确的是 D。

A) 函数的实参和其对应的形参占用相同的存储单元

B) 形参只是形式上的存在, 不占用具体存储单元

C) 同名的实参和形参占同一存储单元

D) 在函数运行时, 函数的形参和实参分别占用不同的存储单元

## 二、填空题 (每小题 2 分, 共 14 分)

1. 已定义“int i = 6, j;”, 执行“j = i || i--”以后, i 的值为 6。

2. 设有定义语句“int a[8] = {1};”, 则数组元素 a[1] 的值为 0。

3. 已有定义“int x = 108, \*p = &x;”, 则 \*p 的值为 108。

4. 设有定义“FILE \*fp;”, 请将以下打开文件的语句补充完整, 以便向一个文本文件 test.txt 中写入内容, 并且当此文件不存在时, 将自动建立新文件。

```
fp = fopen ("test.txt", "w");
```

5. 已有宏定义:

```
#define SUB(x,y) (x)*(y)
```

则宏替换后, `printf("%d", SUB(a++,--b))` 被替换为: `printf("%d", a*(b+1));` (a++)\*(--b)

6. 定义结构类型的关键字是 struct。

7. 将 `int` 型定义为 `Integer` 的语句是 `typedef int Integer;`

三、阅读下面各程序, 写出运行结果 (每小题 4 分, 共 28 分)

1. 有如下程序:

```
#include <stdio.h>
int main(void)
{
    int a = 5, b = 8, c, d;
    c = ++a;
    d = b++;
    a = 2 + 3, 10 / 3;
    b = (2 + 3, 10 / 3);
    printf("%d %d %d %d\n", a, b, c, d);
    return 0;
}
```

运行结果是: 5 3 6 8

2. 有以下程序:

```
#include <stdio.h>
int main(void)
{
    int a = 6, b = 8, x, y;
    x = a < b ? a : b;
    y = (a < b ? a : b);
    printf("%d %d %d %d\n", a, b, x, y);
    return 0;
}
```

运行结果是: 6 8 6 6

3. 有以下程序:

```
#include <stdio.h>
int main(void)
{
    int i = 3;
    do{
        printf("%d ", i);
    }while (--i != 0);
    printf("%d\n", i);
    return 0;
}
```

运行结果是: 3 2 1 0

4. 有以下程序:

```

#include <stdio.h>
int main(void)
{
    int a = 1, b = 5, c, d;
    c = (a % b <= 1) || (a / (++b) < 1);
    d = (a % b <= 1) && (a / (++b) < 1);
    printf("%d %d %d %d\n", a, b, c, d);
    return 0;
}

```

运行结果是: 1 5 1 1

5. 有以下程序:

```

#include <stdio.h>
int f(int n)
{
    if (n == 1) return 3;
    else if (n == 2) return 3;
    else return f(n - 1) + f(n - 2);
}
int main(void)
{
    printf("%d ", f(1));
    printf("%d ", f(2));
    printf("%d ", f(3));
    printf("%d\n", f(4));
    return 0;
}

```

运行结果是: 3 3 6 9

6. 有以下程序:

```

#include <stdio.h>
int main(void)
{
    char s[] = "abcde", *p = s;
    printf("%c %c %c %c\n", *s, s[1], *(p + 2), p[3]);
    return 0;
}

```

运行结果是: a b c d

7. 有以下程序:

```

#include <stdio.h>
#define f(x) x*x
int main(void)
{
    int a, b, c, d;
    a = f(1+1);
    b = f((1+1));
    printf("%d %d ", a, b);
}

```

1+1\*1+1<3

```

        c = f(++a);
        d = f(b--);
        printf("%d %d\n", c, d);
        return 0;
    }

```

运行结果是: ~~4~~ 4 25 10

#### 四、程序填空题（每空 2 分，共 12 分）

1. 下面程序的功能是求一个一维数组各元素的最小值，将程序补充完整。

```

#include<stdio.h>
int Min(int a[], int n)
{
    int i, m = a[0];
    for (i = 1; i < n; i++)
        if (a[i] < m) m = a[i];
    return m;
}
int main(void)
{
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, n = 10;
    printf("%d\n", Min(a, n));
    return 0;
}

```

2. 下面程序的功能是计算一个二维数组各元素之和，将程序补充完整。

```

#include<stdio.h>
int main(void)
{
    int i, j, s = 0;
    int a[3][3] =
    {
        {1, 2, 6},
        {2, 3, -10},
        {-5, 1, 5}
    };
    for (i = 0; i < 3; i++)
        for (j = 0; j < 3; j++)
            s = s + a[i][j];
    printf("和为%d\n", s);
    return 0;
}

```

3. 以下程序的功能是：输出 100 以内（不含 100）能被 3 整除且个位数为 6 的所有正整数，请填空。

```

#include <stdio.h>
int main(void)
{
    int i, j;

```

```

    for (i = 0; 5; i++)
    {
        j = i * 10 + 6;
        if (6) continue;
        printf("%d ", j);
    }
    printf("\n");
    return 0;
}

```

*Handwritten notes:*  $j < 10$  (above the loop condition),  $j \% 3 \neq 0$  (next to the if condition), and a red checkmark.

### 五、编程题（每小题 10 分，共 30 分）

1. 编程实现输出 1~100 之间能被 6 整除但不能被 8 整除的所有整数。
2. 输入一行字符，输出其中包含的数字字符的个数。
3. 从键盘输入一行字符，把这些字符输出到磁盘文本文件 my\_file.txt 中，再从文件 my\_file.txt 中读出这些字符并显示在屏幕上。