

一 选择题和填空题

函数定义和调用

1. 以下正确的函数声明形式是 ()。
A. `int fun(int a; int y)` B. `int fun(int x, int y);` ✓
C. `int fun(int x, int y)` D. `int fun(int x, y);`
2. 函数 `fun((1, 2), (3, 4), (5, 6, 7))` 调用语句中含有参数的个数为 ()。
A. 1 B. 4 C. 2 D. 3 ✓
3. 以下说法正确的是 ()。
~~A. 函数体中可以包含任何 C 语言的语句~~
B. 函数体不可以为空
~~C. 若为空函数则一对大括号也不可以省略~~
~~D. 函数定义时大括号里的叫做函数体~~
4. 以下说法正确的是 ()。
A. 函数声明语句可以放在调用函数的函数体之内也可以在调用函数的函数体语句之外
B. 函数声明语句要求在函数体之外
~~C. 函数声明的目的是告知编译器函数的参数个数、类型与返回值~~
D. 函数的声明语句后面可以省略分号
E. 函数声明的目的是告知编译其函数的参数个数、类型、参数名字和返回值
5. 以下程序正确的说法是 ()。
A. 函数不能相互调用
~~B. 主函数可以调用其他函数~~
~~C. 主调函数和被调函数书写的先后顺序没有要求~~
D. 一个源文件内主调函数必须在被调函数下面
6. C 语言中，函数调用时，以下说法中正确的是 ()
~~A) 实参与其对应的形参各占独立的存储单元;~~
B) 实参与其对应的形参共占同一个存储单元;
C) 只有当实参与其对应的形参同名时，才共占同一个存储单元;
D) 形参是虚拟的，不占存储单元。
7. C 语言规定：简单变量做实参时，它和对应形参之间数据的传递方式为()。
A) 地址传递;
~~B) 单向值传递;~~
C) 由实参传给形参，再由形参传回给实参;
D) 由用户指定传递方式。
8. 以下说法正确的是 ()。
~~A. 实际参数是指函数调用时的参数~~
B. 实际参数是指函数定义时括号内的参数
~~C. 实际参数的类型可以是变量、常量、表达式~~
~~D. 实际参数的个数和类型要求和形参一一对应~~
9. 以下说法错误的是 ()。
A. 如果实参与形参类型不一致，以实参类型为准
~~B. 如果函数值的类型与返回值的类型不一致，以函数类型为准~~

C. return 后边的值不能为表达式

D. 定义函数时，形参的类型说明可以放在函数体内

10. 以下说法正确的是（ ）。

A. 函数调用时参数的传递顺序是从右至左的

B. 实参的计算顺序是从左至右的

C. 实参的计算顺序是从右至左的

☒ D. 实际参数的计算顺序根据编译环境而定

E. 函数调用时参数的传递顺序是从左至右的

11. 假设函数类型为 float，则以下说法错误的是（ ）。

☒ A. 函数是通过 return 语句返回确定值的

B. 有没有 return 语句函数都返回值

C. 函数不能返回值

D. 通过 return 语句可以返回多个值

12. 在 C 语言程序中以下正确的描述是（ ）。

A. 函数的定义可以嵌套，但函数的调用不可以嵌套

☒ B. 函数的定义不可以嵌套，但函数的调用可以嵌套

C. 函数的定义和函数的调用均可以嵌套

D. 函数的定义和函数的调用均不可以嵌套

13. 设有程序段

```
int a[3][4];
```

```
f(a);
```

有以上数组定义和 f 函数调用语句，则在 f 函数的说明中，对形参数组 array 的错误定义方式为（ ）。**列长度不能省**

☒ A. f(int array[3][])

B. f(int array[3][4])

C. f(int array[2][4])

D. f(int array[][4])

14. 设函数有以下定义

```
int fun(int a, int b)
```

```
{
```

```
    return a * b;
```

```
}
```

下列选项中所有变量都已经正确定义并赋值，错误的函数调用是（ ）。

选择一个答案

A. printf("%d", fun(x, y));

☒ B. z = fun(fun(x, y)x, y);

C. z = fun(fun(x, y), fun(x, y));

D.

```
    if ( fun ( x , y ) )
```

```
    {
```

```
        .....
```

```
    }
```

15. 设有程序段

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

```
int main()
```

```
{
```

```

int F = 5, k;
void fun_1();
.....
k = fun_1(F);
.....
return 0;
}

```

以上程序有语法错误，有关错误原因的错误的描述是（ ）。

- ☒ A. 函数说明和函数调用语句之间有矛盾
- ☐ B. 语句 `void fun();` 有错，它是函数调用语句，不能用 `void` 说明
- ☐ C. 变量名不能使用大写字母
- ☐ D. 函数名不能使用下划线

16. 设有程序

```

#include <stdio.h>
int sub(int n)
{
    return (n/10 + n%10);
}
int main()
{
    int x, y;
    scanf("%d", &x);
    y = sub(sub(sub(x)));
    printf("%d\n", y);
    return 0;
}

```

有以上程序，若运行时输入：1234<回车>，程序的输出结果是（ ）。

- ☐ A. 19
- ☐ B. 127
- ☒ C. 10
- ☐ D. 7

17. 设有程序

```

include <stdio.h>
void fun(int x)
{
    printf("%d", x++);
}
int main()
{
    int x = 20;
    fun(++x);
    printf("\n");
    return 0;
}

```

上面程序输出的结果是（ ）。

- ☐ A. 18
- ☐ B. 22
- ☐ C. 20
- ☒ D. 21

18-21函数写法要记住

18. 下面函数实现 `strlen` 函数的功能，即计算指针 `p` 所指向的字符串中的实际字符个数。

```

unsigned int  MyStrlen(char *p)
{
    int  len;
    len = 0;
    for (; *p != '\0'; p++)
    {
        len ++;
    }
    return  len;
}

```

19. 下面函数实现 strcpy 函数的功能，既将 srcStr 内容复制到 dstStr 中，请填空。

```

char *  MyStrcpy(char *dstStr, char  *srcStr)
{
    int  i = 0;
    while (srcStr[i] != '\0')
    {
        dstStr[i] = srcStr[i];
        i++;
    }
    dstStr[i] = '\0';
    return  dstStr;
}

```

20. 下面函数实现 strcat 函数的功能，既将 srcStr 内容链接到 dstStr 后，请填空。

```

char* myStrCat(char *dstStr,char *srcStr)
{
    int i,j;

    for (i = 0;dstStr[i] != '\0';i++); //查找 dstStr 最后一个字符'\0'的位置

    for (j = 0;srcStr[j] != '\0';j++)
    {
        dstStr[i+j] = srcStr[j];
    }
    dstStr[i+j] = '\0'; //加一个空字符

    return dstStr;
}

```

21. 下面函数实现 strcmp 函数的功能，请填写完整

```

int MyStrcmp(char  *s, char  *t)
{
    int i;

    for (i=0; s[i] == t[i]; i++)

```

```

{
    if (s[i] == '\0')
        return 0;
}
return s;
}

```

一维数组做函数形参

22. 若用数组名作为函数调用时的实参，则实际上传递给形参的是：。
- ☒ A) 数组的首地址;
 - ☐ B) 数组的第一个元素值;
 - ☐ C) 数组中全部元素的值;
 - ☐ D) 数组元素的个数。
23. 数组作为函数参数时，下面说法正确的是 ()。
- ☐ A. 传递的是数组的首地址，是双向的"值传递"
 - ☒ B. 传递的是数组的首地址，是"址传递"
 - ☒ C. 传递的是数组的首地址，形参数组和实参数组共占一段内存单元
 - ☒ D. 传递的是数组的首地址，形参数组元素变化后，实参数组元素的值也跟着变化
24. 有以下程序段
- ```

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

```
- 上面程序输出的结果是 ( )。
- ☐ A. 1098765
  - ☐ B. 876543
  - ☒ C. 321678
  - ☐ D. 345678

25. 设有程序
- ```

#include <stdio.h>

```

```

void fun(int *a, int n)
{
    int t, i, j;
    for(i = 0; i < n-1; i++)
        for(j = i+1; j < n; j++)
            if(a[i] < a[j])
            {
                t = a[i];
                a[i] = a[j];
                a[j] = t;
            }
}

int main()
{
    int c[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0}, i;
    fun(c+4, 6);
    for(i = 0; i < 10; i++) printf("%d, ", c[i]);
    printf("\n");
    return 0;
}

```

上面程序输出的结果是（ ）。

- A. 0, 9, 8, 7, 6, 5, 1, 2, 3, 4,
- B. 0, 9, 8, 7, 6, 5, 4, 3, 2, 1,
- C. 1, 2, 3, 4, 5, 6, 7, 8, 9, 0,
- D. 1, 2, 3, 4, 9, 8, 7, 6, 5, 0,

二维数组做函数形参

26. 有以下程序段

```

#include <stdio.h>
#include <string.h>
void fun(char s[][10], int n)
{
    char t;
    int i, j = 0;
    for(i = 0; i < n-1; i++)
        for(j = i+1; j < n; j++)
            if(s[i][0] > s[j][0])
            {
                t = s[i][0];
                s[i][0] = s[j][0];
                s[j][0] = t;
            }
}

```

```

}
int main()
{
    char ss[5][10] = {"bcc", "bbcc", "xy", "aaaacc", "aabcc"};
    fun(ss, 5);
    printf("%s,%s\n", ss[0], ss[4]);
    return 0;
}

```

上面程序输出的结果是 ()。

- ☒ A. acc, xabcc
 ☐ B. aaaacc, xy
 ☐ C. xy, aaaacc
 ☐ D. xcc, aabcc

递归函数

27. 有以下程序段

```

#include <stdio.h>
int fun(int n)
{
    if(n == 1)
        return 1;
    else
        return(n + fun(n-1));
}
int main()
{
    int x;
    scanf("%d", &x);
    x = fun(x);
    printf("%d\n", x);
    return 0;
}

```

有以上程序，执行程序时，给变量 X 输入 10，程序的输出结果是 ()。

- ☐ A. 54
 ☐ B. 65
☒ C. 55
☐ D. 45

28. 有以下程序段

```

#include <stdio.h>
int fun(int a, int b)
{
    if(b == 0)
        return a;
    else
        return(fun(--a, --b));
}
int main()
{
    printf("%d\n", fun(4, 2));
}

```

```
    return 0;
}
```

上面程序输出的结果是 ()。

- A. 3 ~~B. 2~~ C. 1 D. 4

29. 有以下程序段

```
#include <stdio.h>
int f(int x)
{
    int y;
    if(x == 0 || x == 1) return 3;
    y = x * x - f(x-2);
    return y;
}
```

```
int main()
{
    int z;
    z = f(3);
    printf("%d\n", z);
    return 0;
}
```

上面程序输出的结果是 ()。

- A. 8 B. 9 C. 6 D. 0

30. 有以下程序段

```
#include <stdio.h>
long fun5(int n)
{
    long s;
    if((n == 1) || (n == 2))
        s = 2;
    else
        s = n + fun5(n-1);
    return s;
}
```

```
int main()
{
    long x;
    x = fun5(4);
    printf("%ld\n", x);
    return 0;
}
```

上面程序输出的结果是 ()。

- A. 5 ~~B. 9~~ C. 14 D. 2

31. 有以下程序段


```

#include <stdio.h>
int fun(int t[], int n)
{
    int m;
    if(n == 1)
        return t[0];
    else
        if(n >= 2)
        {
            m = fun(t, n-1);
            return m;
        }
}
int main()
{
    int a[] = {11, 4, 6, 3, 8, 2, 3, 5, 9, 2};
    printf("%d\n", fun(a, 10));
    return 0;
}

```

上面程序输出的结果是 ()。

- A. 7 B. 6 C. 11 D. 9

32. 设有程序段

```

#include <stdio.h>
int foo(int n);

int main()
{
    printf("%d\n", foo(2.5));
}

```

```

int foo(int n)
{
    if (n<=0)
        return 0;
    else
        return n+foo(n-1);
}

```

该程序的运行结果为: 3

变量的生存期和作用域、存储类别、类型限定

33. 下面说法正确的是 ()。

- A. 局部变量的有效范围从定义处到文件结束
B. 在函数体外定义的变量一定不是局部变量

- ☒ C. 局部变量可以是动态的也可以是静态的
D. 局部变量只能是在函数体内定义的变量
34. 在一个 C 源程序文件中所定义的全局变量，其作用域为（ ）。
A. 所在程序的全部范围
B. 所在函数的全部范围
☒ C. 由具体定义位置和 `extern` 说明来决定范围
D. 所在文件的全部范围
35. 下面说法正确的是（ ）。
A. 全局变量可以是动态的也可以是静态的
☒ B. 形式参数是局部变量
☒ C. 在不同函数中可以使用相同名字的变量
☒ D. 在函数内定义的变量只在本函数范围内有效
36. 若在一个函数中的复合语句中定义了一个变量，则关于该变量下面说法错误的是（ ）。
A. 在本程序范围内均有效 ☒ B. 只在复合语句中有效
C. 在该函数中有效 D. 为非法变量
37. C 语言中形参的缺省存储类别是（ ）。
☒ A) 自动(auto) B) 静态(static) C) 寄存器(register) D) 外部(extern)
38. 凡是函数中未指定存储类别的局部变量，其隐含的存储类别为（ ）。
☒ A. auto B. extern C. register D. static
39. 在一个 C 程序中若要定义一个只在本文件内使用的全局变量，应该使用的存储类别是（ ）。
☒ A. static B. register C. auto D. extern
40. 如果要定义一个只允许在本文件函数调用的函数应该使用的关键字是（ ）。
A. register ☒ B. static C. extern D. auto

41. 设有程序

```
#include <stdio.h>
int fun(int x)
{
    static int t = 0;
    return t += x;
}
int main()
{
    int s, i;
    for(i = 1; i <= 5; i++)
        s = fun(i);
    printf("%d\n", s);
    return 0;
}
```

上面程序输出的结果是（ ）。

- A. 6 ☒ B. 15 C. 10 D. 3
42. 设有程序
- ```
#include <stdio.h>
int f(int a)
{
```

```
static c = 3;
a = c++;
return a;
}
int main()
{
 int a = 2, i;
 static int k = 0;
 for(i = 0; i < 2; i++)
 k += f(i);

 printf("%d\n", k);
 return 0;
}
```

上面程序输出的结果是 ( )。

A. 5

B. 6

C. 7

D. 4

## 二、程序题目

以下是函数这章练习题目

课后作业 3、4

程序作业 第 17 次作业的 2、3；第 16 次作业的 1；第 14、15 次作业全部

实验手册：第 8 章实验 1、2