

函数

1 什么是函数

函数 = 完成某一特定功能的代码模块

把一段功能明确、可重复使用的代码封装起来

使用函数的好处：

- 降低代码重复
- 提高可读性
- 便于调试与维护
- 支持模块化程序设计
- 大型程序的基本单位

2 函数三要素

1-函数名

2-参数列表（可有可无）

3-返回值类型（必须有，哪怕是 void）

3 函数的基本结构

```
返回值类型 函数名(参数类型 参数名, ...) {  
    函数体;  
    return 返回值;  
}
```

两数求和

示例



```
int add(int a, int b) {  
    return a + b;  
}
```

4 函数的声明与定义

1-函数声明

```
int add(int, int);
```

作用：

- 告诉编译器 函数的存在、参数和返回类型
- 解决“**先用后定义**”的问题

2-函数定义

```
int add(int a, int b) {  
    return a + b;  
}
```

3-函数使用

```
int c = add(3, 5);
```

5 函数的返回值

1-有返回值的函数

```
int max(int a, int b) {  
    return a > b ? a : b;  
}
```

特点：

- **return** 返回一个值
- 函数执行到 **return** 立即结束

2-没有返回值的函数 (void)

```
void printHello() {  
|     printf("Hello\n");  
| }  
|
```

void 表示不返回任何值

Return作用总结:

1-return 表达式; --返回值并结束函数

2-return; --结束函数 (用于 void)

6 函数参数

1-形参与实参

形参是局部变量

```
int add(int a, int b); // a, b: 形参  
add(3, 5);           // 3, 5: 实参
```

2-值传递

传递的是值的拷贝

```
✓ void func(int x) {  
|     x = 10;  
| }  
int a = 5;  
✓ func(a);  
| //a=5 值不会改变
```

3-指针传递

函数操作的是实参的地址

```
|  
void func(int *x) {  
|     *x = 10;  
| }  
int a = 5;  
func(&a);  
| //a=10
```

7

函数与数组

数组传参 == 指针传参

```
void printArr(int arr[], int n) {  
    for (int i = 0; i < n; i++)  
        printf("%d ", arr[i]);  
}
```



等价

```
void printArr(int *arr, int n)
```

传数组长度的原因：

1-数组名退化为指针

2-无法直接得到数组大小

8

函数与字符串：字符串参数本质也是指针

```
void printStr(char *s) {  
    while (*s) {  
        putchar(*s++);  
    }  
}
```

9

递归函数

递归函数：自己调用自己

三要素：

递归出口（终止条件）

递归关系（递推公式）

参数变化，趋近出口

10 局部变量 & 全局变量（函数作用域）

1-局部变量

```
void f() {  
    int x = 10;  
}
```

作用域：函数内部

生命周期：函数调用期间

2-全局变量

```
int g = 10;
```

```
void f() {  
    g++;  
}
```

特点：

整个程序可见

生命周期贯穿程序始终

1 1 函数的嵌套与调用关系

C 语言不允许函数定义嵌套

```
void f1() {  
    void f2() { } // ✗ 错误  
}
```

但允许函数调用嵌套

```
f1(f2());
```



例题

1-函数递归--斐波那契数列

数学定义:

$$F(n) = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F(n-1) + F(n-2), & n \geq 2 \end{cases}$$

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

```
int fib(int n)
{
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return fib(n - 1) + fib(n - 2);
}
```

```
int main()
{
    int n;
    scanf("%d", &n);
    printf("%d\n", fib(n));
    return 0;
}
```

```
fib(4)
├─ fib(3)
│   ├─ fib(2)
│   │   ├─ fib(1) → 1
│   │   └─ fib(0) → 0
│   └─ fib(1) → 1
└─ fib(2)
    ├─ fib(1) → 1
    └─ fib(0) → 0
```

C 语言中规定函数的返回值的类型是由 D 。

- A. 调用该函数时系统临时决定
- B. 调用该函数时的主调用函数类型所决定
- C. `return` 语句中的表达式类型所决定
- D. 在定义该函数时所指定的函数类型所决定

若用数组名作为函数调用时实参，则实际上传递给形参的是 D 。

- A. 数组元素的个数
- B. 数组中全部元素的值
- C. 数组的第一个元素值
- D. 数组首地址

以下对 C 语言函数的有关描述中，正确的是 D 。

- A. 函数必须有返回值，否则不能使用函数
- B. 函数既可以嵌套定义又可以嵌套调用
- 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
- D. 调用传值函数时，只能把实参的值传送给形参，形参的值不能传送给实参

以下错误的描述是 B 。

- A. 形式参数是从属于函数的局部变量
- B. 在一个函数内部的复合语句中定义的变量可以在本函数范围内有效
- C. 一个函数内部定义的变量只能在本函数范围内有效
- D. 不同的函数中可以定义名字相同的变量

10、对于 C 语言的函数，下列叙述中正确的是 (A)

- A. 函数定义不能嵌套，但函数调用可以嵌套
- B. 函数定义可以嵌套，但函数调用不能嵌套
- C. 函数定义和函数调用都不能嵌套
- D. 函数定义和函数调用都可以嵌套