
《C 语言开发入门教程》习题答案

第 1 章 初识 C 语言

一、填空题

- 1、汇编语言、高级语言
- 2、.c
- 3、Stadio.h
- 4、return
- 5、编译、链接

二、判断题

- 1、×
- 2、√
- 3、×
- 4、√
- 5、√

三、选择题

- 1、A
- 2、B
- 3、C、D
- 4、D
- 5、B

四、简答题

1、`printf()`，该函数用于输出一行信息，可以简单理解为向控制台输出文字或符号等。`printf()`括号中的内容称为函数的参数，括号内可以看到输出的字符串“Hello, world\n”，其中“\n”表示换行操作，它不会输出到控制台。

2、注释是对程序的某个功能或者某行代码的解释说明，它只在 C 语言源文件中有效，在编译时会被编译器忽略。

五、编程题

1、参考答案

```
#include<stdio.h>
void main()
{
    printf("我喜欢 C 语言!");
    getchar(); //等待输入一个字符，起到让运行结果界面暂停
}
```

第 2 章 数据类型与运算符

一、填空题

- 1、0、0x/0X
- 2、下划线
- 3、原码、反码、补码
- 4、构造类型、空类型
- 5、获取数据字节数

二、判断题

- 1、×
- 2、×
- 3、√
- 4、×
- 5、√

三、选择题

- 1、A、B、D
- 2、B
- 3、B
- 4、C
- 5、D

四、简答题

1、在 C 语言中根据运算符进行运算需根据变量的个数可以将运算符分为单目运算符、双目运算符和三目运算符，其区别与示例如下表：

名称	运算所需变量个数	范例
单目运算符	1 个	++、--、!、sizeof、~、...
双目运算符	2 个	+、-、*、/、%、<<、==、...
三目运算符	3 个	?:

2、在进行自增(++)和自减(--)的运算时，如果运算符(++或--)放在操作数的前面则是先进行自增或自减运算，再进行其他运算。反之，如果运算符放在操作数的后面则是先进行其他运算再进行自增或自减运算。

五、编程题

1、参考答案

```
#include<stdio.h>
void main()
{
    float a = 1.5, b = 2.6, h = 5;
    float s;
    s = (a + b)*h;
    printf("梯形的面积为: %.2f\n", s); //%.2f 输出小数点后 2 位数字
    getchar();
}
```

2、参考答案

```
#include<stdio.h>
void main()
{
    int a = 3, b = 4;
    a = a^b;
    b = b^a;
    a = a^b;
    printf("a = %d, b = %d\n", a, b);
    getchar();
}
```

第 3 章 流程控制

一、填空题

- 1、break
- 2、顺序
- 3、循环
- 4、if、if...else、if...else if...else
- 5、3

二、判断题

- 1、× 2、× 3、√ 4、× 5、√

三、选择题

- 1、C 2、A、C 3、B、C、D 4、D 5、A

四、简答题

1、跳转语句

- break 当它出现在 switch 条件语句中时，作用是终止某个 case 并跳出 switch 结构。当它出现在循环语句中，作用是跳出当前循环语句，执行后面的代码
- continue 在循环语句中，立即终止本次循环，并执行下一次循环。
- goto 语句可以跳出外层循环，需要对外层循环添加标记，然后使用 goto 语句可以跳转到标记的语句处。

2、for 与 while 的相同之处都是完成一定次数的循环；其不同之处是：while 一般用于未知次数的循环；for 一般用于次数已知的循环；

两者声明变量的作用域不同，如果定义一个变量 i，在 while 循环后 i 的值会存在，但 for 循环后，i 的值不存在了；

五、编程题

1、参考答案

```
#include<stdio.h>
void main()
{
    int x = 5;
    if (x % 2 == 0)
        printf("%d 为偶数\n",x);
    else
        printf("%d 为奇数\n",x);
    getchar();
}
```

2、参考答案

```
#include<stdio.h>
void main()
{
    int i, sum = 0;
    for (i = 0; i < 100; i++) //控制 i 在 1-100 之间
    {
        if (i % 2 != 0) //如果是奇数
            sum += i; //就加起来
    }
    printf("sum = %d\n", sum);
    getchar();
}
```

第 4 章 函数

一、填空题

- 1、return
- 2、有参、无参
- 3、scanf()
- 4、递归
- 5、全局、局部

二、判断题

1、√ 2、× 3、× 4、√ 5、√

三、选择题

1、B 2、D 3、A、B、C 4、B、C 5、A、B、C

四、简答题

1、

● 局部变量

局部变量就是在函数内部声明的变量，它只在本函数内有效，也就是说，只能在本函数内使用它。此外，局部变量只有当它所在的函数被调用时才会被使用，而当函数调用结束时局部变量就会失去作用。

● 全局变量

全局变量(包括 main()函数)，它不属于哪一个函数，而是属于源程序。因此全局变量可以为程序中的所有函数所共用。它的有效范围从定义处开始到源程序结束。

2、

● 外部函数

可以被其他源文件调用的函数称为外部函数，在调用外部函数之前，需要在当前源文件中定义外部函数。定义外部函数的方式是在函数的返回值类型前面添加 **extern** 关键字。

● 内部函数

函数只在它的定义文件中有效，该类函数称为内部函数。在定义内部函数时，需要在函数的返回值类型前面添加 **static** 关键字（又称为静态函数）。它不可以被别的源文件调用，只在当前源文件中有效。

五、编程题

1、参考答案

```
#include<stdio.h>
int _sum(int i)
{
    if (i == 1)
        return 1;
    if (i > 1)
        return i + _sum(i - 1); //函数自身调用自身
}
void main()
{
    int sum;
    sum = _sum(10);
    printf("sum = %d\n", sum);
}
```

```
    getchchar();  
}
```

2、参考答案

```
#define _CRT_SECURE_NO_WARNINGS //scanf()函数是不安全的，调用它要关闭  
安全检查
```

```
#include<stdio.h>  
#include<stdlib.h>  
int sum(int a,int b)  
{  
    return a + b;  
}  
void main()  
{  
    int a,b,he;  
    scanf("%d%d", &a,&b); //用户从键盘输入数据  
  
    he = sum(a, b);  
  
    printf("he = %d\n", he);  
    system("pause"); //包含在stdlib.h头文件中，也可以让运行结果界面暂停  
}
```

第5章 数组

一、填空题

- 1、集合
- 2、首
- 3、变量
- 4、下标
- 5、9

二、判断题

- 1、√ 2、× 3、× 4、× 5、√

三、选择题

- 1、D 2、A 3、C 4、A、C 5、A、B、C、D

四、简答题

1、一维数组指的是只有一个下标的数组，其定义方式如下所示：

```
类型说明符 数组名[常量表达式];
```

其初始化有三种方法：

- 直接对数组中的所有元素赋值；如 `int i[5]={1,2,3,4,5};`
- 只对数组中的一部分元素赋值；如 `int i[5]={1,2,3};`
- 对数组全部元素赋值，但不指定长度；如 `int i[]={1,2,3,4};`

2、一维数组和二维数组区别如下

- 一维数组只有一个下标，二维数组有两个下标；
- 二维数组有行与列之分，一维数组则没有；
- 赋值时一维数组下标可以略，二维数组可以省略第一个下标，第二个下标不能省略。

五、编程题

1、参考答案

```
#include<stdio.h>
#include<stdlib.h>
int _max(int a[5]) //数组作为函数参数
{
    int i;
    int max = a[0];
    for (i = 1; i < 5; i++)
    {
        if (max < a[i])
            max = a[i];
    }
    return max;
}
void main()
{
    int a[5] = { 3, 4, 6, 9, 13 };
    printf("max = %d\n", _max(a));
    system("pause");
}
```

2、参考答案

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

```
void _sort(int a[], int n)
{
    int i, j, temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = 0; j < n - i - 1; j++)
        {
            if (a[j] > a[j + 1])
            {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
            }
        }
    }
}

void main()
{
    int i;
    int b[] = { 25, 24, 12, 76, 101, 96, 28 };
    printf("排序前: ");
    for (i = 0; i < 7; i++)
        printf("%d ", b[i]);

    _sort(b, 7); //数组名作为函数参数

    printf("\n排序后: ");
    for (i = 0; i < 7; i++)
        printf("%d ", b[i]);

    system("pause");
}
```


第 6 章 指针

一、填空题

- 1、指针
- 2、字符数
- 3、&
- 4、函数指针
- 5、二级指针

二、判断题

- 1、√ 2、√ 3、× 4、√ 5、×

三、选择题

- 1、C 2、A 3、B 4、C 5、A

四、简答题

- 1、在程序运行过程中，数据都是保存在内存中的，内存是以字节为单位的连续存储空间，每个字节都有一个编号，这个编号称为地址。变量也是有地址的，专门用于存放其他变量的地址，这种变量称为指针变量，通常简称指针。通过指针可以操作它指向的内存中的变量。取内存的地址或取内存中的值进行运算。如 `char arr[10], *p; p = arr;` `arr` 是大小为 10 的指针数组，`p` 是指向这个数组的指针。
- 2、数组指针是指向数组的指针，数组中存储的是统一的其他数据类型的变量；指针指向这一块内存的首地址。可以通过指针的移动来依次访问数组中的元素。
指针数组是指数组中存储的是指针，数组中指针的数据类型必须一致，指针数组的定义方式与普通数组类似，只需在类型前面加上“*”，如 `int *arr[10]`；存储了 10 个 `int` 类型的指针。

五、编程题

1、参考答案

```
#define _CRT_SECURE_NO_WARNINGS //scanf()函数是不安全的，调用它要关闭安全检查
#include<stdio.h>
#include<stdlib.h>
void swap(int *a, int *b)
{
    int temp ;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

```
}  
void main()  
{  
    int a, b;  
    printf("请输入两个数: \n");  
    scanf("%d%d", &a, &b);  
    swap(&a, &b);  
    printf("交换后: \n%d %d\n", a, b);  
    system("pause");  
}
```

2、参考答案

```
#include<stdio.h>  
#include<stdlib.h>  
void copystr(char *form1, char *form2)  
{  
    while (*form1)  
    {  
        *form2 = *form1;  
        form2++;  
        form1++;  
    }  
    *form2 = '\0';  
}  
void main()  
{  
    char str1[20] = "Hello World";  
    char str2[20] = "I am Chinese";  
    printf("复制前: \nstr1:%s\nstr2:%s\n", str1, str2);  
    copystr(str1, str2);  
    printf("复制后: \nstr1:%s\nstr2:%s\n", str1, str2);  
    system("pause");  
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/946025121052010105>