

**【教材】**《C 语言程序设计（第 2 版）》清华大学出版社，黄保和，江弋 编著。2011 年 10 月第二版。ISBN:978-7-302-26972-4。售价：35 元。

**【答案版本】**本习题答案为 2012 年 2 月修订版本。

### 一、思考题

**1、函数总需要从 main 中调用吗？当调用一个函数时，为什么要使用参数？**

函数不是总需要从 main 函数中调用，使用参数的目的是为了给被调函数传递数据。

**2、什么是函数的返回值？是否每个函数都有返回值？为什么？对它们调用有什么不同？**

函数返回值是被调函数传递给主调函数的数据，不是每个函数都需要有返回值，是否有返回值由函数的功能确定，可以将有返回值的函数赋值给同类型的变量，无返回值得则不可。

**3、每个函数的调用都需要有参数吗？为什么？**

不是，对于参数函数的调用不需要参数。

**4、C 编译器提供的库函数与你自己编写的函数有什么不同？在调用它们时，应注意什么？**

系统将所有库函数的函数原型分成多组，每组放在一个头文件中。调用库函数时，要注意引用其对应的头文件。

**5、局部变量和全局变量有何不同？编写程序时，何时使用局部变量或全局变量？**

如果变量定义在一个函数或复合语句的内部，则称该变量为局部变量，局部变量的作用域是函数级的或者模块级的，如果变量定义在所有函数外部，则称该变量为全局变量，其作用域是从它的定义的位置开始到本源文件结束。选择变量类型的两个原则是：1 当变量只在某函数或复合语句内使用时，不要定义成全局变量。2 当多个函数都引用同一个变量时，在这些函数上面定义全局变量，而且定义部分尽量靠近这些函数。

**6、静态局部变量和静态全局变量有何异同？何时应该使用它们？**

静态局部变量是在函数或复合语句内部用 static 声明的变量，静态静态局部变量存储静态存储区，且仅在编译时赋值一次。静态全局变量是使用 static 声明的全局变量，静态全局变量属于静态存储类型，编译时将静态全局变量分配在静态存储区，其生存期是程序的整个运行期间，但他的作用域只限于本文件。

**7、指出下列程序中的错误何在？**

```
Void do_it_first(){
    printf( "This is first" );
    void main();
    return;
}
main(){
    printf( "This is second" );
    return 0;
}
```

- 1、void 写成了 Void
  - 2、主函数应该叫 main，而不是 do\_it\_first
  - 3、main 函数只能有系统调用
  - 4、程序最开始执行的一定是 main 函数，任何函数不可能在 main 函数之前执行
- 修改后：

```
#include<stdio.h>
int Fun()
{
    printf("This is second");
    return 0;
}
void main()
{
    printf("This is first");
    Fun();
    return ;
}
```

二、选择题

1、C语言程序由函数组成。它的主函数可以在其它函数之后，函数内不可以嵌套定义函数。

- A.主函数必须在其他函数之前，函数内可以嵌套定义函数
- B.主函数可以在其他函数之后，函数内不可以嵌套函数
- C.主函数必须在其他函数之前，函数内不可以嵌套定义函数
- D.主函数必须在其他函数之后，函数内可以嵌套定义函数

2、一个C语言程序的基本组成单位是C) 函数

- A.主程序
- B.子程序
- C.函数
- D.过程

3、以下说法正确的是C) C语言程序总是从主函数main()开始执行

- A.C 语言程序总是从第一个定义的函数开始执行
- B.在 C 语言程序中，要调用的函数必须在 main（）函数中定义
- C.C 语言程序是从主函数 main（）开始执行
- D.C 语言程序中的 main（）函数必须放在程序的开始部分

4、已知函数 fun 的定义为

```
void fun（）{…}
```

则函数定义中的void的含义是A) 执行函数fun后，函数没有返回值

- A.执行函数fun后，函数没有返回值
- B.执行函数fun后，可以返回任意类型
- C.执行函数fun后，函数不再返回
- D.以上三个答案都是错误的

5、下列对C语言函数的描述中，正确的是A) 在C语言中，调用函数时只能将实参的值传递给形参，形参的值不能传递给实参。

- A.在 C 语言中，调用函数时只能将实参的值传递给形参，形参的值不能传递给实参。
- B.函数必须有返回值，否则不能使用函数
- C.C 语言函数既可以嵌套定义，又可以递归调用
- D.C 程序中有调用关系的所有函数都必须放在同一个源程序文件中

6、以下叙述中错误的是B) 在C语言中，外部变量的缺省值存储类别是自动的。

- A. 在 C 语言中，函数形参的存储类型是自动（auto）类型的变量

- B. 在 C 语言中，外部变量的缺省值存储类别是自动的。
- C. 在 C 语言中，在调用函数时，实参和对应形参在类型上只需赋值兼容
- D. 在 C 语言中，函数中的自动变量可以赋初值，每调用一次赋初值一次

**7、C语言中的函数D) 嵌套调用和递归调用均可**

- A.不可以嵌套调用
- B.可以嵌套调用，但不能递归调用
- C.可以嵌套定义
- D.嵌套调用和递归调用均可

**8、C语言中函数返回值r的类型是由D) 定义函数时所指定的函数类型**

- A.调用该函数的主调函数类型
- B.函数参数类型
- C.return 语句中的表达式类型
- D.定义函数时所指定的函数类型

**9、C语言规定，调用一个函数时，实参变量与形参变量之间的数据传递方式是D) 值传递**

- A.由实参传给形参，并由形参传回来给实参
- B.地址传递
- C.由用户指定传递方式
- D.值传递

**10、下列的结论中正确的是B) 所有的递归程序均可以采用非递归算法实现**

- A.所有的递归程序均不可以采用非递归算法实现
- B.所有的递归程序均可以采用非递归算法实现
- C.只有部分递归程序可以用非递归算法实现
- D.以上三种说法都不对

**11、在下列结论中，只有一个是错误的，它是B) 有些递归程序是不能用非递归算法实现的**

- A.C 语言中允许函数的递归调用
- B.有些递归程序是不能用非递归算法实现的
- C.C 语言允许函数的递归调用
- D.C 语言中 continue 语句，可以通过改变程序的结构而省略

**12、下列叙述错误的是C) 主函数中定义的变量在整个程序中都是有效的**

- A.形参是局部变量
- B.复合语句中定义的变量只在该复合语句中有效
- C.主函数中定义的变量在整个程序中都是有效的
- D.其他函数中定义的变量在主函数中不能使用

**13、若函数类型和return语句中的表达式类型不一致，则B) 不会出错，且返回值的类型以函数类型为准**

- A.运行时出现不确定结果
- B.返回值的类型以函数类型为准
- C.编译时出错
- D.返回值的类型以 return 语句中表达式的类型为准

14、下面函数定义正确的是 **C) double fun(float u, float v){return u+v;}**

- A. double fun (double u, v){ return u + v; }
- B. double fun (double u; double v){ return u + v; }
- C. double fun (float u, float v){ return u + v; }
- D. double fun (u, v){ float u, v; return u + v; }

15、以下函数 fun 的返回值类型为 **C) int**

**fun(double x){ printf(“%f\n”,x);}**

- A.double
- B.void
- C.int
- D.无法确定类型

### 三、编程题

1、写一个判别素数的函数，在主函数输入一个整数，输出其是否为素数的信息。

```
#include<stdio.h>
int sushu(int);
main()
{
    int a;
    printf("输入一个整数: ");
    scanf("%d",&a);
    if(sushu(a)==1)
        printf("是素数");
    else
        printf("不是素数");
}
int sushu(int x)
{
    int i;
    for(i=2;i<x;i++)
    {
        if(x%i==0)
            return 0;
    }
    return 1;
}
```

2、写一个判别水仙花数的函数，求出所有的水仙花数。

说明：所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数本身。例如：153 是一个“水仙花数”，因为  $153=1$  的三次方+ $5$  的三次方+ $3$  的三次方。

```
#include<stdio.h>
int shuixian(int);
main()
{
    int i;
    for(i=100;i<1000;i++)
    {
```

```

        if(shuixian(i)==1)
            printf("%d\t",i);
    }
}
int shuixian(int x)
{
    int i,sum=0,a;
    a=x;
    for(i=0;i<3;i++)
    {
        sum+=(x%10)*(x%10)*(x%10);
        x/=10;
    }
    if(sum==a)
        return 1;
    else
        return 0;
}

```

3、编写函数求两个整数的最小值，在主函数中通过键盘输入  $x$  和  $y$  的值，调用该函数，输出最小值。

```

#include <stdio.h>
int min( int x, int y)
{
    return x < y ? x : y;
}

void main()
{
    int x, y;
    printf("请输入两个整数:\n");
    scanf("%d%d", &x, &y);
    printf("两个数中较小的一个是: %d\n ", min( x, y));
}

```

4、编写函数对  $x$  进行判断，如果  $x$  的值大于 0，返回 1；否则返回 0，在主函数中通过键盘输入  $x$  的值，调用该函数，输出返回值。

```

#include <stdio.h>
int Fun( int x)
{
    if ( x > 0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

```

```

}

void main()
{
    int x;
    printf("请输入一个整数:\n");
    scanf("%d", &x);
    printf("Fun 函数的返回值是: %d\n ", Fun(x));
}

```

5、利用随机函数 rand()计算圆周率的近似值

```

#include <math.h>
#include <time.h>
using namespace std;
double calPI(int n)
{
    int x,y;
    int r=1000;
    int k=0;
    for (int i=0;i<n;i++)
    {
        x = rand()% r+1;
        y = rand()% r+1;
        if (x*x+y*y<=1000000)
        {
            k++;
        }
    }
    return k*4.0/n;
}
int main()
{
    int n;
    srand(time(NULL));
    while(1)
    {
        scanf("%d",&n);
        if(n<=0) break;
        printf("PI=%f\n",calPI(n));
    }
    return 0;
}

```

6、写一个函数，其功能是返回正整数 n 的位数。并设计一个主程序测试这个函数。

```

#include<stdio.h>
int weishu(int x)
{

```

```

int i,c=0;
for(i=0;x>0;i++)
{
    x/=10;
    c++;
}
return c;
}
main()
{
    int a;
    printf("输入一个整数: ");
    scanf("%d",&a);
    printf("共有%d 位。",weishu(a));
}
    
```

7、写两个函数，分别求两个整数的最大公约数和最小公倍数，用主函数调用这两个函数，并输出结果，两个整数由键盘键入。

```

int gcd(int m,int n)
{
    if (m < n)
    {
        int tmp = m;
        m = n;
        n = tmp;
    }
    if (n == 0)
        return m;
    while (n > 0)
    {
        int tmp = m % n;
        m = n;
        n = tmp;
    }
    return m;
}

int gbs(int partA,int partB)//最小公倍数
{
    return partA*partB/gcd(partA,partB);
}
    
```

8、编写函数，验证陈景润研究的哥德巴赫猜想：任意大偶数为两个素数之和，并输出这两个素数（所谓大偶数是指 6 开始的偶数）

说明：下面的程序，需要输入查询范围的左右边界 a,b，然后程序会验证属于(a,b)区间内的所有偶数都是两个素数之和，并且会打印出这些素数。

```

#include<stdio.h>
int sushu(int x)
    
```

```

{
    int i;
    for(i=2;i<x;i++)
    {
        if(x%i==0)
            return 0;
    }
    return 1;
}
main()
{
    int i,j,a,b;
    printf("输入查询范围: (两个偶数)");
    scanf("%d%d",&a,&b);
    for(i=a;i<=b;i+=2)
    {
        printf("%d:\n",i);
        for(j=2;j<=i/2;j++)
        {
            if(sushu(j)&& sushu(i-j))
                printf("%d+%d=%d\n",j,i-j,i);
        }
        printf("\n");
    }
}

```

9、利用递归算法，编写一程序，计算组合数  $C(m,n)$ (提示：利用  $C(m,n)=C(m-1,n)+C(m-1,n-1)$ )

```

int comb(int m,int n)
{
    if(m==0||n==0||m<=n)
    {
        return 1;
    }
    if(n==1)
    {
        return m;
    }
    else
    {
        return comb(m-1,n)+comb(m-1,n-1);
    }
}

```

10、用递归算法，求两个正整数  $m$ 、 $n$  的最大公约数

```

int gcd(int parA, int parB)//最大公约数
{
    if (parB == 0)
    {

```

```

        return parA;
    }
    else
    {

        return gcd(parB, parA % parB);
    }
}

```

11、请分别用递归和非递归的算法，编写一个函数，其功能是计算  $n^k$  的值（其中  $n$  和  $k$  是两个整数）。并用你设计的函数显示一张  $3^k$  的表， $k$  从 0-9

```

int exponent(int n,int k)//递归
{
    if(k==1)
    {
        return n;
    }
    if(k==0)
    {
        return 1;
    }
    else
    {
        return n*exponent(n,k-1);
    }
}

int exponent2(int n,int k)//非递归
{
    int result=1;
    for(int i=1;i<=k;i++)
    {
        result=result*n;
    }
    return result;
}

void printExponent()//打印
{
    for(int i =0;i<=9;i++)
    {
        printf("%8d ",exponent2(3,i));
        if((i+1)%3==0)
            printf("\n");
    }
}

```

12.编程设计一个简单的猜数游戏：由随机函数产生1—100之间的整数，请你猜，如果你猜对了，则显示“你猜对了！”，游戏结束。否则显示“你猜错了！”，并告诉你所猜的数是太大了还是太小了，直到你猜对为止。程序统计你才对所用的次数，并输出。

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

void main()
{
    int n = 0;
    int nNumber = 0;
    int yourInput = 1;
    srand(time(NULL));
    n = rand() % 101 ;
    printf("请输入一个整数:\n");
    scanf("%d", &yourInput);
    while( yourInput != n )
    {
        if ( yourInput > n )
        {
            printf("\n 你猜错了! \t 太大了\t 请重新输入:");
        }
        else
        {
            printf("\n 你猜错了! \t 太小了\t 请重新输入:");
        }
        nNumber++;
        scanf("%d", &yourInput);
    }
    printf("你猜对了! \t 共猜了%d 次\n ", nNumber);
}
```

=====此后无内容=====