

第 3 章



函 数

思考题

1. 什么是函数？如何进行函数定义？函数声明与定义有什么不同？

参考答案：

在 C++ 程序中，函数就是程序的基本模块，即将代码逻辑或算法编成一个个相对独立的代码块，通过组合函数模块的调用实现程序特定的功能。函数的声明与函数的定义形式上十分相似，但是二者有着本质上的不同。声明是不开辟内存的，仅告诉编译器，要声明的部分存在，要预留一点空间。定义则需要开辟内存。

2. 什么是函数的返回类型？通过什么方式传递返回值？

参考答案：

函数的返回类型即函数返回值的类型，应当在定义函数时指定该类型。函数的返回值通过 `return` 语句返回给主调函数。

3. 形式参数与实际参数分别是什么？阐明二者的区别和联系。

参考答案：

函数的参数分为形参和实参两种。在定义函数时，函数名后面括号中的变量名称为“形式参数”，简称“形参”，在主调函数中调用函数时，函数后面括号中的参数称为“实际参数”，简称“实参”。

形参出现在函数定义中，作用范围是整个函数体，离开该函数则不能使用。实参出现在主调函数中，进入被调函数后，实参变量也不起作用。形参和实参的功能是用来进行数

据传输。发生函数调用时,主调函数把实参的值传送给被调函数的形参,从而实现主调函数向被调函数的数据传输。

4. 如何调用一个已经定义的函数? 参数是如何传递的?

参考答案:

根据函数调用在程序中出现的位置,函数调用方式可以分为以下三种。

函数表达式:函数作为表达式中的一项出现在表达式中,以函数返回值参与表达式的运算。这种方式要求函数是有返回值的。

函数语句:函数调用的一般形式加上分号即构成函数语句。

函数实参:函数作为另一个函数调用的实际参数出现。这种情况是把该函数的返回值作为实参进行传送,因此要求该函数必须是有返回值的。

5. 局部变量和全局变量有什么区别和联系?

参考答案:

局部变量也称为内部变量。局部变量是在函数内做定义说明的,其作用域仅限于函数内,离开该函数后不能再使用这种变量。

全局变量也称为外部变量,它是在函数外部定义的变量。它不属于哪一个函数,而是属于整个源程序文件,其有效范围为从定义变量的位置开始到本源文件结束。全局变量提供了函数间数据联系的渠道。

6. 比较值调用和引用调用的相同点与不同点。

参考答案:

相同点:都是用来向函数传递参数的方法,都可以用来传递基本数据类型和自定义数据类型(如结构体、类)。

不同点:值调用传递的是参数的副本,而引用调用传递的是参数的引用(地址)。在值调用中,对参数的任何修改都不会影响到原始数据,而在引用调用中,对参数的修改会影响到原始数据。值调用会产生参数的副本,可能会增加内存消耗和执行时间,而引用调用则不会产生额外的副本,效率更高。对于指针参数,引用调用可以避免空指针的问题,而值调用无法做到。

7. 函数原型中的参数名与函数定义中的参数名以及函数调用中的参数名是否必须一致。

参考答案:

不必一致。

8. 简述什么是嵌套调用和递归调用。

参考答案:

嵌套调用是指在一个函数内部调用另一个函数,也可以是在被调用的函数内部再调

用其他函数,形成多层次的调用关系。嵌套调用可以用来组织和重用代码,使程序结构更清晰、更模块化。

递归调用是指在函数内部直接或间接地调用自身的过程。递归调用通常包含一个递归出口(基本情况),用于终止递归调用,以避免无限循环。递归调用常用于解决可以被分解为相同问题的子问题,或者涉及树结构、分支结构等自相似性质的问题。递归调用是一种简洁而优雅的编程技术,但需要谨慎设计,避免栈溢出和效率低下的问题。

练习题

1. 计算函数 $F(x, y, z) = (x+z)/(y-z) + (y+2 \times z)/(x-2 \times z)$ 的值,要求将 F 函数实现为子函数,由主函数向子函数传参调用。

参考答案:

问题分析

函数 $F(x, y, z)$ 需要按照子函数语法要求定义,而计算涉及三个变量 x, y, z ,需要从键盘接收,而计算过程有除法操作,需要进行除数不为 0 的判断,可以定义子函数 `isValid` 来判断除数是否正当。当输入合理时,输出相应结果;否则,给出提示。

编写程序

```
1  #include <iostream>
2  using namespace std;
3  int F(int x, int y, int z)
4  {
5      int result = (x+z)/(y-z) + (y+2*z)/(x-2*z);
6      return result;
7  }
8  bool isValid(int x, int y, int z)
9  {
10     if(y-z==0)
11         return false;
12     if(x-2*z==0)
13         return false;
14     return true;
15 }
16 int main()
17 {
18     int x = 0;
19     int y = 0;
20     int z = 0;
21     cout <<"请分别输入参数 x y z,以空格隔开: ";
22     cin >> x >> y >> z;
23     if(isValid(x,y,z))
24         cout <<"经 F 函数计算后,结果为:" << F(x,y,z);
25     else
26         cout <<"您的输入参数将使得除数为 0,无法计算";
```

```
27     return 0;
28 }
```

运行结果：

请分别输入参数 x y z, 以空格隔开: 1 2 3
经 F 函数计算后, 结果为: -5

2. 要求用户输入一个年月日, 判断用户输入的日期是否合法。请定义子函数实现合法性验证。

参考答案：

问题分析

日期的合法性运算过程涉及三个变量, 分别是年(year)、月(mon)、日(day)。按照要求定义子函数 isLegal 进行判断, 首先需要保证年、月、日处于合理的区间, 接下来需要对具体月份进行分析, 有 31 天的月份和 30 天的月份需要分类别分析, 而二月份的判断需要结合闰年与否进行。

编写程序

```
1  #include <iostream>
2  using namespace std;
3  bool isLeapYear(int year)
4  {
5      if(((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0))
6          return true;
7      return false;
8  }
9
10 bool isLegal(int year, int mon, int day)
11 {
12
13     if(year < 0 || mon <= 0 || mon > 12 || day <= 0 || day > 31)
14         return false;
15     if(1 == mon || 3 == mon || 5 == mon || 7 == mon ||
16         8 == mon || 10 == mon || 12 == mon)
17     {
18         return true;
19     }
20     if(isLeapYear(year))
21     {
22         if(2 == mon && (28 == day || 30 == day || 31 == day))
23             return false;
24         return true;
25     }
26     else
27     {
28         if(2 == mon && (29 == day || 30 == day || 31 == day))
29             return false;
```

```
30         return true;
31     }
32 }
33 int main()
34 {
35     int year = 0;
36     int mon = 0;
37     int day = 0;
38     cout << "请分别输入年月日,以空格隔开:\n ";
39     cin >> year >> mon >> day;
40     if(isLegal(year, mon, day))
41         cout << "合法";
42     else
43         cout << "不合法";
44     return 0;
45 }
```

运行结果:

```
请分别输入年月日,以空格隔开:
1992 10 32↵
不合法
```

3. 银行要受理客户的贷款请求,请设计程序要求根据客户的贷款额度、期限和利率,计算并输出客户最终需要的还款总额。

参考答案:

问题分析

客户还款总额 totalPay 的计算与三个变量有关,分别是贷款额度 loanSum、期限 year 和利率 annualRate。运算过程需要用到幂函数操作,可以从 math.h 文件中引用,而不需要自己实现。

编写程序

```
1 #include <iostream>
2 #include <math.h>
3 using namespace std;
4 int main()
5 {
6     int year = 0;           //贷款年数
7     double annualRate = 0; //年利率
8     double loanSum = 0;    //贷款总额
9     double totalPay = 0;   //总支付额
10    cout << "请输入年贷款利率:";
11    cin >> annualRate;
12    cout << "请输入贷款年限:";
13    cin >> year;
```

```
14     cout << "请输入贷款总额:";
15     cin >> loanSum;
16     totalPay = loanSum * pow(1 + annualRate, year);
17     cout << "一共需要还款:" << totalPay << endl;
18     return 0;
19 }
```

运行结果:

```
请输入年贷款利率: 0.10↵
请输入贷款年限: 2↵
请输入贷款总额: 10000↵
一共需要还款: 12100
```

4. 我国的个人所得税按照累进税计算,要求用户输入其收入,输出其应缴纳的个人所得税。

扣税基数: 3500 元,计算公式: 个人所得税额 = 每月应纳税额 × 税率 - 速算扣除数

全月应纳税所得额(已减扣税基数)	税率(%)	速算扣除数
小于或等于 0 元	0	0
不超过 1500 元	3	0
超过 1500 元至 4500 元的部分	10	105
超过 4500 元至 9000 元的部分	20	555
超过 9000 元至 35 000 元的部分	25	1005
超过 35 000 元至 55 000 元的部分	30	2755
超过 55 000 元至 80 000 元的部分	35	5505
超过 80 000 元的部分	45	13 505

参考答案:

问题分析

个人所得税 tax 的计算与个人的月收入 salary、缴费基数和税率相关。当收入低于缴费基数时,不需要缴纳;当收入高于缴费基数时,需要具体判断收入位于哪个缴费区间,采取累进方式进行计算。

编写程序

```
#include <iostream>
using namespace std;
int main( )
{
    double salary, tax = 0;
    double value;
    cout << "请输入您本月的收入:";
    cin >> salary;
    value = salary - 3500;
    if(value <= 0.0)
```

```

        tax = 0.0;
    else
    {
        if(value <= 1500)
            tax = value * 0.03 - 0.0;
        else if(value <= 4500)
            tax = value * 0.10 - 105.0;
        else if(value <= 9000)
            tax = value * 0.20 - 555.0;
        else if(value <= 35000)
            tax = value * 0.25 - 1005.0;
        else if(value <= 55000)
            tax = value * 0.30 - 2755.0;
        else if(value <= 80000)
            tax = value * 0.35 - 5505.0;
        else
            tax = value * 0.45 - 13505.0;
    }
    cout << "您本月应缴个人所得税" << tax << "元";
    return 0;
}

```

5. 编写函数计算三角形的面积,公式如下。

$$s = \frac{1}{2}(a + b + c)$$

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

三角形的三边 a 、 b 、 c 的值由主函数输入。

参考答案:

问题分析

定义 calculateTriangleArea 函数接收三角形的三边长作为参数,根据题目给出的公式计算并返回三角形的面积。在主函数中,用户输入三角形的三边长,并调用函数计算并输出面积。

编写程序

```

#include <iostream>
#include <cmath>
using namespace std;

//计算三角形面积的函数
double calculateTriangleArea(double a, double b, double c) {
    //计算半周长
    double s = (a + b + c) / 2.0;
    //计算面积
    double area = sqrt(s * (s - a) * (s - b) * (s - c));
    return area;
}

```

```
}

int main() {
    double a, b, c;
    //输入三角形的三边长
    cout << "请输入三角形的三条边长(a,b,c):";
    cin >> a >> b >> c;
    //调用函数计算面积
    double area = calculateTriangleArea(a, b, c);
    //输出结果
    cout << "三角形的面积为:" << area << endl;
    return 0;
}
```

6. 编写函数,使用递归法求 $f(x) = \sum_{i=0}^n i^2$, n 的值由主函数输入。

参考答案:

问题分析

利用递归的方式,不断将问题分解为更小的子问题。首先确定递归出口,当 n 等于 0 时,平方和为 0。然后,在递归的过程中,每次计算前 $n-1$ 项的平方和,并加上 n^2 ,直到递归到 n 等于 0 时返回结果。这样,就能够逐步累加得到 $f(x)$ 的值。

编写程序

```
#include <iostream>
using namespace std;

//递归函数,计算 f(x)
int sumOfSquares(int n) {
    if (n == 0)
        return 0;
    //递归调用,计算前 n-1 项的平方和,然后加上 n^2
    return sumOfSquares(n - 1) + n * n;
}

int main() {
    int n;
    cout << "请输入一个正整数 n:";
    cin >> n;
    //调用递归函数计算 f(x)
    int result = sumOfSquares(n);
    cout << "f(" << n << ") = " << result << endl;
    return 0;
}
```

7. 编写函数:统计字符串中字母的个数、数字的个数、空格的个数、其他字符的个数。主函数中从键盘输入一行字符,并输出统计结果。

参考答案:

问题分析

遍历输入的字符串,逐个检查每个字符的类型,并统计不同类型字符的个数。首先,初始化各个计数器为0。然后,遍历字符串的每个字符,使用 `isalpha()`、`isdigit()` 和 `isspace()` 函数判断字符的类型,分别对字母、数字和空格进行计数。最后,对于不属于以上三种类型的字符,视为其他字符并进行计数。最终,将各个类型字符的计数结果输出。

编写程序

```
#include <iostream>
#include <cctype>
using namespace std;
//函数:统计字符串中字母、数字、空格和其他字符的个数
void countCharacters(const string& str, int& letters, int& digits, int& spaces, int& others) {
    letters = 0;
    digits = 0;
    spaces = 0;
    others = 0;
    for (char ch : str) { //遍历字符串的每个字符
        //使用 cctype 头文件中的函数判断字符的类型
        if (isalpha(ch)) {
            letters++;
        } else if (isdigit(ch)) {
            digits++;
        } else if (isspace(ch)) {
            spaces++;
        } else {
            others++;
        }
    }
}

int main() {
    string input;
    cout << "请输入一行字符:";
    getline(cin, input);
    int letterCount, digitCount, spaceCount, otherCount;
    //调用函数统计字符个数
    countCharacters(input, letterCount, digitCount, spaceCount, otherCount);
    cout << "字母个数:" << letterCount << endl;
    cout << "数字个数:" << digitCount << endl;
    cout << "空格个数:" << spaceCount << endl;
    cout << "其他字符个数:" << otherCount << endl;
    return 0;
}
```

8. 所谓“回文”是一种特殊的数或文字短语。它无论顺着读还是倒着读,结果都一样。例如,以下几个6位整数都是回文数:222222,522225,352235,662266。以下5位整数都是回文数:12321,77777,89998和44744。编写函数,输入一个6位或7位整数,判断它是否是回文数。如果是回文数,输出“Yes”;如果不是回文数,输出“No”。整数输入和输出是否为回文数,在主函数中完成。

例如:

请输入一个6位或7位整数:349943

Yes

请输入一个6位或7位整数:3499432

No

参考答案:

问题分析

将输入的整数转换为字符串,并从字符串的两端开始逐个比较字符是否相等,直到中间位置。如果每对对应位置的字符都相等,则整数是回文数,否则不是。

编写程序

```
#include <iostream>
#include <string>

using namespace std;

//函数:判断一个整数是否为回文数
bool isPalindrome(int num) {
    string strNum = to_string(num);    //将整数转换为字符串
    int n = strNum.length();
    for (int i = 0; i < n / 2; ++i) {
        if (strNum[i] != strNum[n - i - 1]) {
            return false;            //如果对应位置的字符不相等,返回 false
        }
    }
    return true;                    //如果全部对应位置的字符都相等,返回 true
}

int main() {
    int num;
    cout << "请输入一个6位或7位整数:";
    cin >> num;
    if (num >= 100000 && num <= 999999) {            //判断是否为6位数
        if (isPalindrome(num)) {
            cout << "Yes" << endl;
        } else {
            cout << "No" << endl;
        }
    } else if (num >= 1000000 && num <= 9999999) {    //判断是否为7位数
```