

编号: 2-1



山东师范大学
SHANDONG NORMAL UNIVERSITY

信息科学与工程学院实验报告

《面向对象程序设计》

Object-Oriented Programming

姓名:	张泽浩
学号:	202111000212
班级:	计工本 2102
导师:	张庆科
时间:	2021 年 10 月 08 日



《面向对象程序设计》实验报告

基本要求：实验报告包含实验目的、实验内容、实验过程（详细操作流程）、实验结果（程序运行结果高清截图）、实验分析总结五个部分。报告中若涉及代码程序，请在附录部分提供完整程序源码及源码托管地址(基于 Highlight 软件导入源码)。报告撰写完毕后请提交 PDF 格式版本报告到课程云班课系统。

一、实验目的

1. 掌握使用 C++ 面向过程程序设计方法
2. 掌握 C++ 对 C 语言的典型改进和扩展
3. 掌握 Visual Studio 建立项目步骤
4. 掌握 Visual Studio 程序调试方法

二、实验内容

（一）任务一：C++ 素数判断

1. 任务要求

建立 VS 项目，在源文件 main.cpp 中定义素数判别函数 `isPrimeNumber()`，在主函数中输入一个整数 `m`，通过布尔类型方式输出该整数是否为素数的信息。

2. 测试案例

```
01 -----
02 请输入任意一个整数：17
03 -----
04 整数17是素数：True
```

（二）任务二：C++ 验证哥德巴赫猜想

1. 任务要求

哥德巴赫猜想：对于任意一个不小于 6 的偶数，均可以将其表示为两个素数之和，例如：6=3+3，8=3+5，12=5+7，.....等等。

请在主函数中输入一个不小于 6 的整数 `N`，然后调用偶数解码函数 `DecodeEvenToPrime(int num)`，输出小于整数 `N` 的所有偶数的素数分解形式。

2. 测试案例

```
01 -----
02 请输入一个整数N (N>6)：20
03 -----
04 偶数20的素数分解结果：20 = 3 + 17
05 偶数20的素数分解结果：20 = 7 + 13
06 偶数18的素数分解结果：18 = 5 + 13
07 偶数18的素数分解结果：18 = 7 + 11
```



- 08 偶数16的素数分解结果: $16 = 3 + 13$
- 09 偶数16的素数分解结果: $16 = 5 + 11$
- 10 偶数14的素数分解结果: $14 = 3 + 11$
- 11 偶数14的素数分解结果: $14 = 7 + 7$
- 12 偶数12的素数分解结果: $12 = 5 + 7$
- 13 偶数10的素数分解结果: $10 = 3 + 7$
- 14 偶数10的素数分解结果: $10 = 5 + 5$
- 15 偶数8的素数分解结果: $8 = 3 + 5$
- 16 偶数6的素数分解结果: $6 = 3 + 3$
- 17 -----
- 18 输出完毕!

三、实验过程

(一) 任务一: C++素数判断

1. 程序流程图

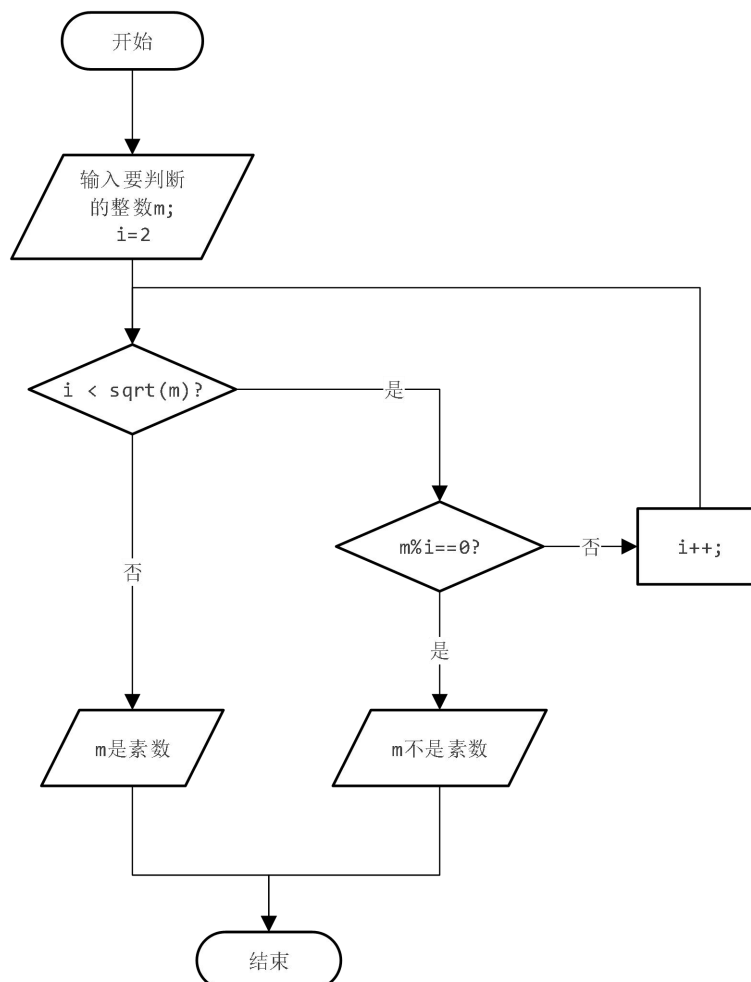


图 1 C++素数判断流程图

2. 实验原理

(1) 素数的定义: 素数, 又叫质数, 是指“大于1的整数中, 只能被1和这个数本身整除的数”。



(2) 判断方法：通过 for 循环，判断所求数 m 能否整除从 2 到 \sqrt{m} 的数据。在程序中，需要引入 `cmath` 的头文件，使用其中的 `sqrt()` 函数求 \sqrt{m} 。如果对于所有大于 2 小于 \sqrt{m} 的数都不是 m 的因数，则 m 是素数；如果对于所有大于 2 小于 \sqrt{m} 的数存在 m 的因数，则 m 是合数。

3. 相关代码

```
// 素数判别函数
bool isPrimeNumber(int m) {
    int top = sqrt(m);
    for (int i = 2; i <= top; i++) {
        if (m % i == 0) {
            return false;
        }
    }
    return true;
}
```

(二) 任务二：C++验证哥德巴赫猜想

1. 程序流程图

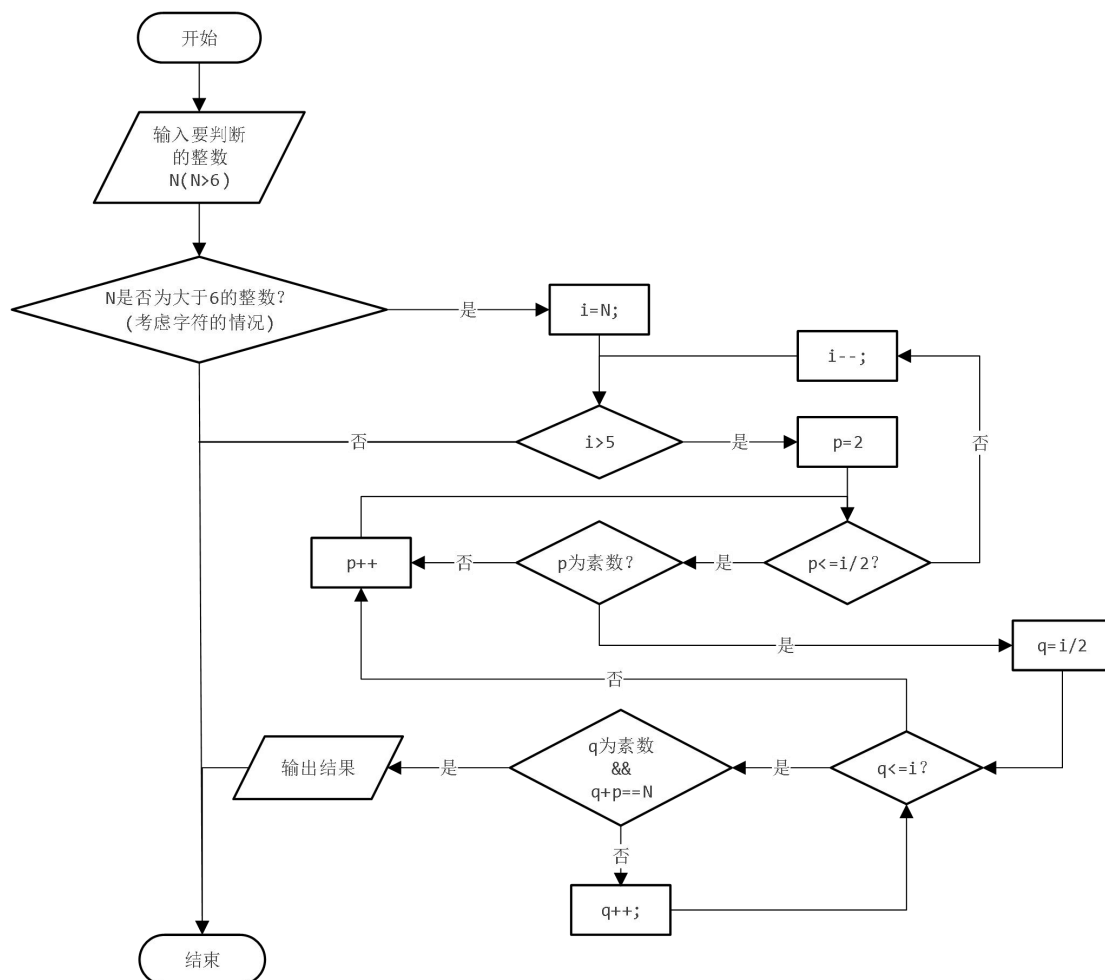


图 2 C++验证哥德巴赫猜想流程图



2. 实验原理

(1) 哥德巴赫猜想: 对于任意一个不小于 6 的偶数, 均可以将其表示为两个素数之和的形式。

(2) 输入数据的判断: 定义一个函数 `cin_to_N()`, 对输入的数据进行判断, 其判断内容包括以下两个方面: 是否整数 (排除字符的影响)、是否为大于 6 的整数。在判断过程中, 用到了字符串转数字的函数 `atoi()`。

(3) 验证哥德巴赫猜想: 通过 `for` 循环遍历小于输入数据 `N` 且大于 5 的数, 在其为偶数的前提下, 判断是否为两个素数的和。判断方式则是将小于 `N` 且大于 5 的数据分为两个区间, 利用 `for` 循环分别遍历判断是否为素数且和为 `N`。

3. 相关代码

```
// 哥德巴赫猜想验证输入函数
int cin_to_N() {
    string temp_str_N;
    int temp_N;
    cout << "-----" << endl;
    cout << "请输入一个整数N (N>6): ";
    cin >> temp_str_N;
    cout << "-----" << endl;
    temp_N = s2i(temp_str_N);
    if (temp_N > 6 && isDigit(temp_str_N)) {
        return temp_N;
    }
    else {
        cout << "请按要求输入数据!" << endl;
        return cin_to_N();
    }
}

// 验证哥德巴赫猜想
void DecodeEvenToPrime(int num) {
    for (int i = num; i > 5; i--) {
        if (i % 2 == 0) {
            for (int p = 2; p <= i / 2; p++) {
                if (isPrimeNumber(p)) {
                    for (int q = i / 2; q <= i; q++) {
                        if (isPrimeNumber(q) && p + q == i) {
                            cout << "偶数" << i << "的素数分解结果: " << i << " = "
                                << p << " + " << q << endl;
                        }
                    }
                }
            }
        }
    }
}
```



四、实验结果

(一) 任务一：C++素数判断

运行结果：

```
D:\C & C++\Object-oriented Programming\Object-oriented Programming\x64\Debug\Experiment-1th.exe
----- 任务一：C++素数判断 -----
请输入任意一个整数：17
整数17是素数：True
请按任意键继续. . .
```

(二) 任务二：C++验证哥德巴赫猜想

1. 错误案例：

(1) 输入字符型数据

```
D:\C & C++\Object-oriented Programming\Object-oriented Programming\x64\Debug\Experiment-1th.exe
----- 任务一：C++素数判断 -----
请输入任意一个整数：17
整数17是素数：True
请按任意键继续. . .

----- 任务二：C++验证哥德巴赫猜想 -----
请输入一个整数N (N>6)：a
请按任意键继续. . .
请输入一个整数N (N>6)：
```

(2) 输入小于 6 的数据



```
D:\C & C++\Object-oriented Programming\Object-oriented Programming\x64\Debug\Experiment-1th.exe

----- 任务一: C++素数判断 -----
请输入任意一个整数: 17
整数17是素数: True
请按任意键继续. . .

----- 任务二: C++验证哥德巴赫猜想 -----
请输入一个整数N (N>6): a
请按任意键继续. . .
请输入一个整数N (N>6): 4
请按任意键继续. . .
请输入一个整数N (N>6):
```

2. 正确案例:

```
D:\C & C++\Object-oriented Programming\Object-oriented Programming\x64\Debug\Experiment-1th.exe

请输入一个整数N (N>6): a
请按任意键继续. . .
请输入一个整数N (N>6): 4
请按任意键继续. . .
请输入一个整数N (N>6): 20
偶数20的素数分解结果: 20 = 3 + 17
偶数20的素数分解结果: 20 = 7 + 13
偶数18的素数分解结果: 18 = 5 + 13
偶数18的素数分解结果: 18 = 7 + 11
偶数16的素数分解结果: 16 = 3 + 13
偶数16的素数分解结果: 16 = 5 + 11
偶数14的素数分解结果: 14 = 3 + 11
偶数14的素数分解结果: 14 = 7 + 7
偶数12的素数分解结果: 12 = 5 + 7
偶数10的素数分解结果: 10 = 3 + 7
偶数10的素数分解结果: 10 = 5 + 5
偶数8的素数分解结果: 8 = 3 + 5
偶数6的素数分解结果: 6 = 3 + 3
输出完毕!
请按任意键继续. . .
```

五、实验总结

通过本次实验，复习了 `cmath` 头文件中的一些函数。在实验过程中，通过搜集资料了解学习了 C++ 中字符串转化为数字的函数 `atoi()`，了解了其基本用法：`int atoi(const char *str)` 把参数 `str` 所指向的字符串转换为一个 `int` 型整数。

在本次实验中，还体会到了将程序段包装成函数的优点，不仅使程序段的结构清晰，也方便程序的跳转(`cin_to_N()`函数完美体现)。



附录：实验源代码（基于 Highlight 软件粘贴带有行号的源码）

代码托管地址：

https://gitee.com/IHRD/object-oriented-programming/tree/master/OOP_Experiment/Experiment-2_1

```
01 #include <iostream>
02 #include <cmath>
03 #include <string.h>
04 using namespace std;
05
06 // 素数判别函数
07 bool isPrimeNumber(int m) {
08     int top = sqrt(m);
09     for (int i = 2; i <= top; i++) {
10         if (m % i == 0) {
11             return false;
12         }
13     }
14     return true;
15 }
16
17 // 将字符串转化为数字
18 int s2i(string s) {
19     return atoi(s.c_str());
20 }
21
22 // 判断是否为数字
23 bool isDigit(string s) {
24     int n = s.size();
25     for (int i = 0; i < n; i++) {
26         if (s[i] - '0' >= 0 && s[i] - '0' <= 9) {
27             continue;
28         }
29         return false;
30     }
31     return true;
32 }
33
34 // 哥德巴赫猜想验证输入函数
35 int cin_to_N() {
36     string temp_str_N;
37     int temp_N;
38     cout << "-----" << endl;
39     cout << "请输入一个整数N (N>6): ";
40     cin >> temp_str_N;
```




```
41     cout << "-----" << endl;
42     temp_N = s2i(temp_str_N);
43     if (temp_N > 6 && isDigit(temp_str_N)) {
44         return temp_N;
45     } else {
46         cout << "请按要求输入数据!" << endl;
47         return cin_to_N();
48     }
49 }
50
51 // 验证哥德巴赫猜想
52 void DecodeEvenToPrime(int num) {
53     for (int i = num; i > 5; i--) {
54         if (i % 2 == 0) {
55             for (int p = 2; p <= i / 2; p++) {
56                 if (isPrimeNumber(p)) {
57                     for (int q = i / 2; q <= i; q++) {
58                         if (isPrimeNumber(q) && p + q == i) {
59                             cout << "偶数" << i <<
60                                 "的素数分解结果: " << i <<
61                                 " = " << p << " + " << q << endl;
62                         }
63                     }
64                 }
65             }
66         }
67     }
68 }
69
70 // 主函数
71 int main() {
72
73     /***** 任务一: C++素数判断 *****/
74     cout << "-----" << endl;
75     cout << "----- 任务一: C++素数判断 -----" <<
76     endl;
77     int m;
78     cout << "-----" << endl;
79     cout << "请输入任意一个整数: ";
80     cin >> m;
81     cout << "-----" << endl;
82     if (isPrimeNumber(m)) {
83         cout << "整数" << m << "是素数: " << "True" <<
84         endl;
```



```
85     } else {
86         cout << "整数" << m << "是素数: " << "False" <<
87         endl;
88     }
89     system("pause");
90     /***** 任务二: C++验证哥德巴赫猜想 *****/
91     cout << endl;
92     cout << "-----" << endl;
93     cout << "- 任务二: C++验证哥德巴赫猜想 -" <<
94     endl;
95     int N = cin_to_N();
96     DecodeEvenToPrime(N);
97     cout << "-----" << endl;
98     cout << "输出完毕! " << endl;
99     system("pause");
100     return 0;
101 }
```