

第六次实验报告

实验环境

以下所有实验都处于这一环境中

操作系统

Windows 10 家庭中文版 64位 版本10.0.17134.345

硬件

CPU Intel Core i7-8750H

RAM 8GB

IDE

Microsoft Visual Studio Community 2017 VisualStudio.15.Release/15.8.5+28010.2036

Visual C++ 2017 00369-60000-00001-AA380

第一题：孪生素数

实验目的

求10到200之间的孪生素数对。孪生素数对指两个素数的值相差2的一对素数。

要求：输出时，每行输出一对，其中小数在前，大数在后。

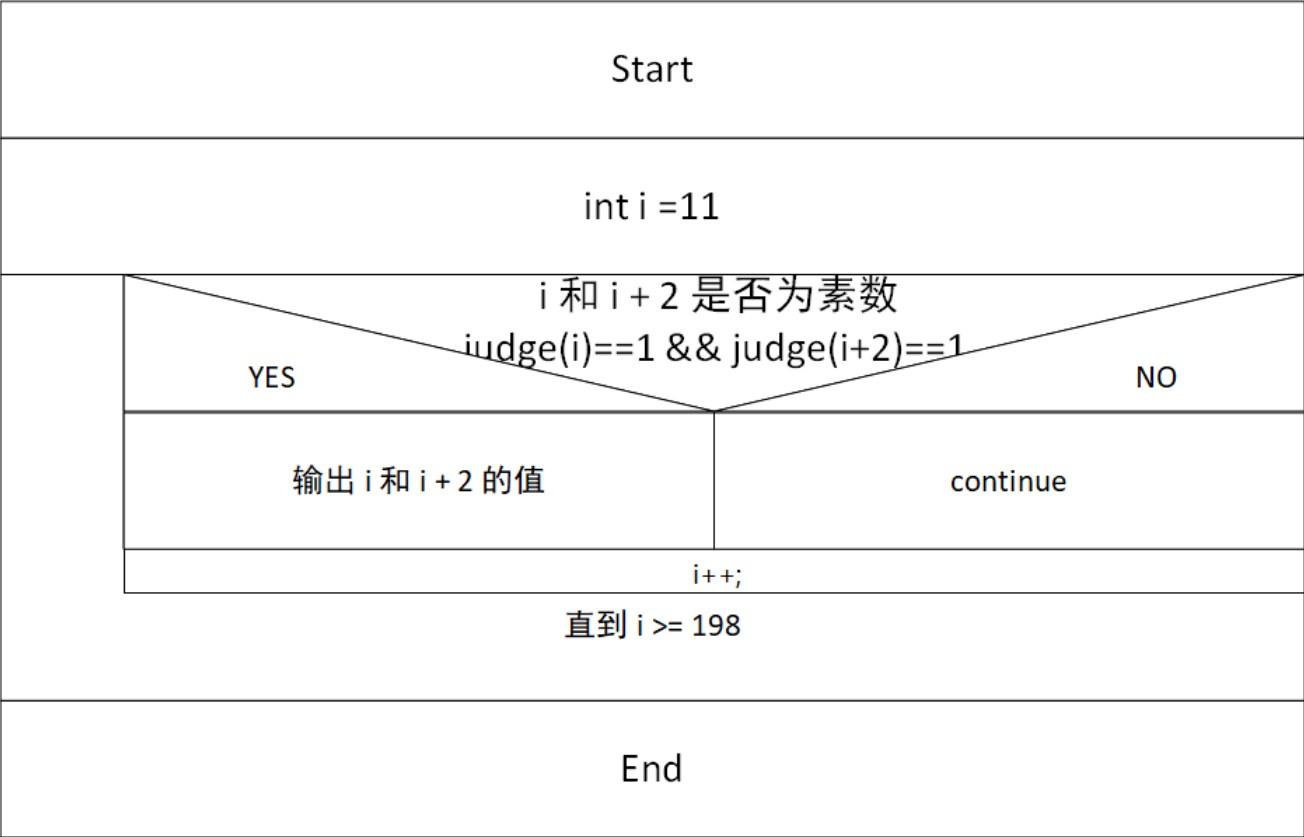
练习循环语句和条件判断

实验内容

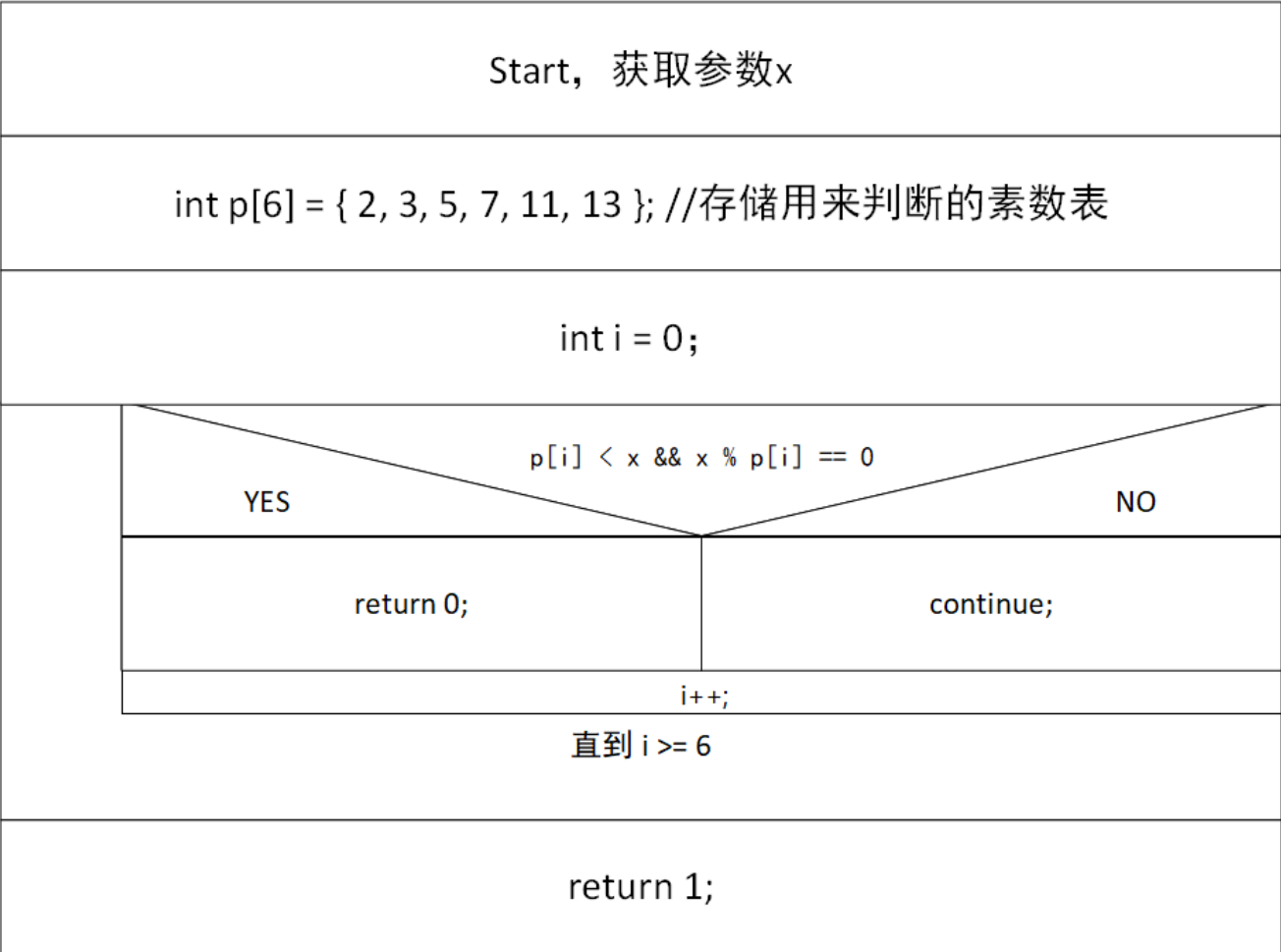
分析

由素数的定义，寻找孪生素数对只需分别判断 n 和 $n+2$ 是否只含有其本身和1两个因数即可。由于数对最大为(198,200)，只需检验 n 和 $n+2$ 能否被 2,3,5,7,11,13 整除即可，而完全由比13大的素数作为因子构成的合数最小为225。而10,198为合数，所以只需测试 $10 < n < 198$ 即可。

所以可得逻辑如下图：



判断 i 和 i + 2 是否为素数的 bool 函数 judge(int x)

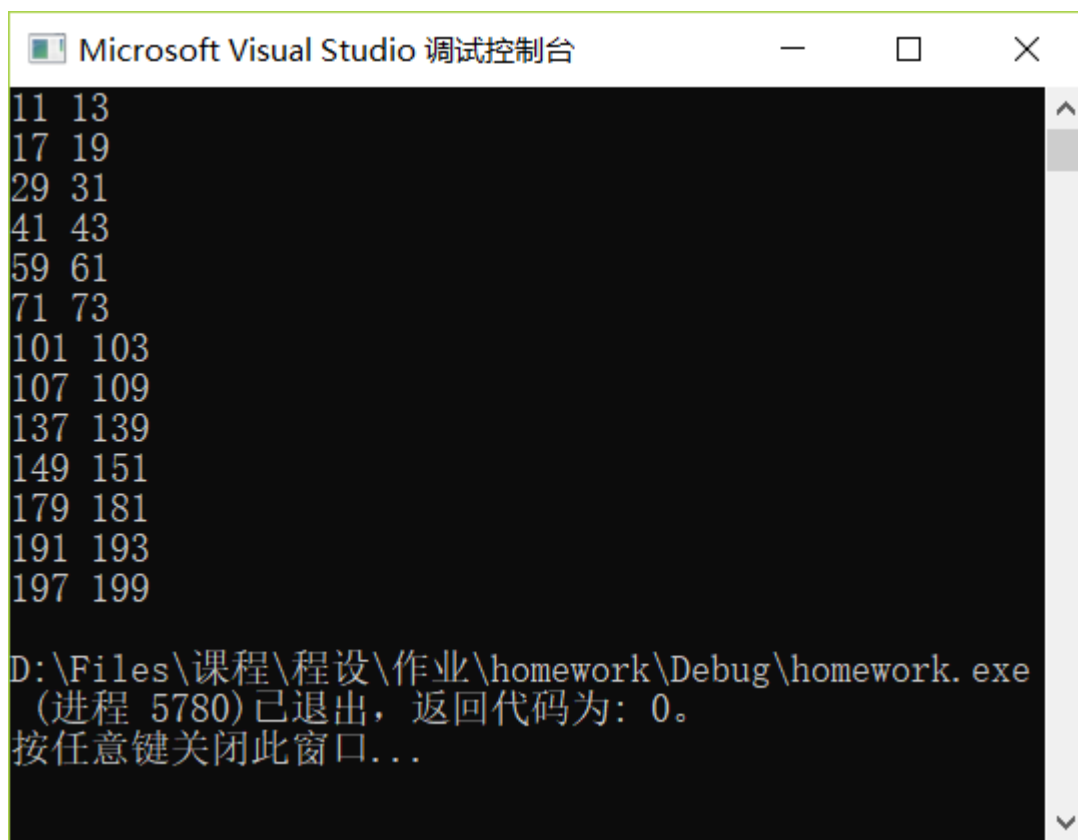


End

代码

```
1  #include <iostream>
2  using namespace std;
3
4  bool judge(int x);
5
6  int main() {
7      for (int i = 11; i < 198; i++) {
8          if (judge(i) && judge(i + 2)) {
9              cout << i << " " << i + 2 << endl;
10             }
11         }
12
13         return 0;
14     }
15
16     bool judge(int x) {
17         int p[6] = { 2, 3, 5, 7, 11, 13 };
18         for (int i = 0; i < 6; i++) {
19             if (p[i] < x && x % p[i] == 0) return 0;
20         }
21         return 1;
22     }
```

结果



```
Microsoft Visual Studio 调试控制台

11 13
17 19
29 31
41 43
59 61
71 73
101 103
107 109
137 139
149 151
179 181
191 193
197 199

D:\Files\课程\程设\作业\homework\Debug\homework.exe
(进程 5780) 已退出, 返回代码为: 0。
按任意键关闭此窗口...
```

分析总结

通过定义函数 `bool judge(int x)` 来判断一个数是否为素数, 使得 `main` 函数里判断 `i` 和 `i + 2` 是否为素数的语句更加清晰。

第二题：菱形打印

实验目的

编写程序, 打印菱形。输入一个整数 n , 若 n 为奇数且 $n > 2$, 打印对角线长度为 n 的菱形。若 n 不为大于 2 的奇数则输出 `Input Error`。即:

- 输入: 一个大于 2 的奇数 n
- 输出: 菱形或 `Input Error`

输入输出示例:

```
1 input n: 8
2 output:
3 Input Error
4
5 input n: 7
6 output:
7     *
8     ***
9     *****
10    *******
11     *****
12     ***
13     *
```

实验内容

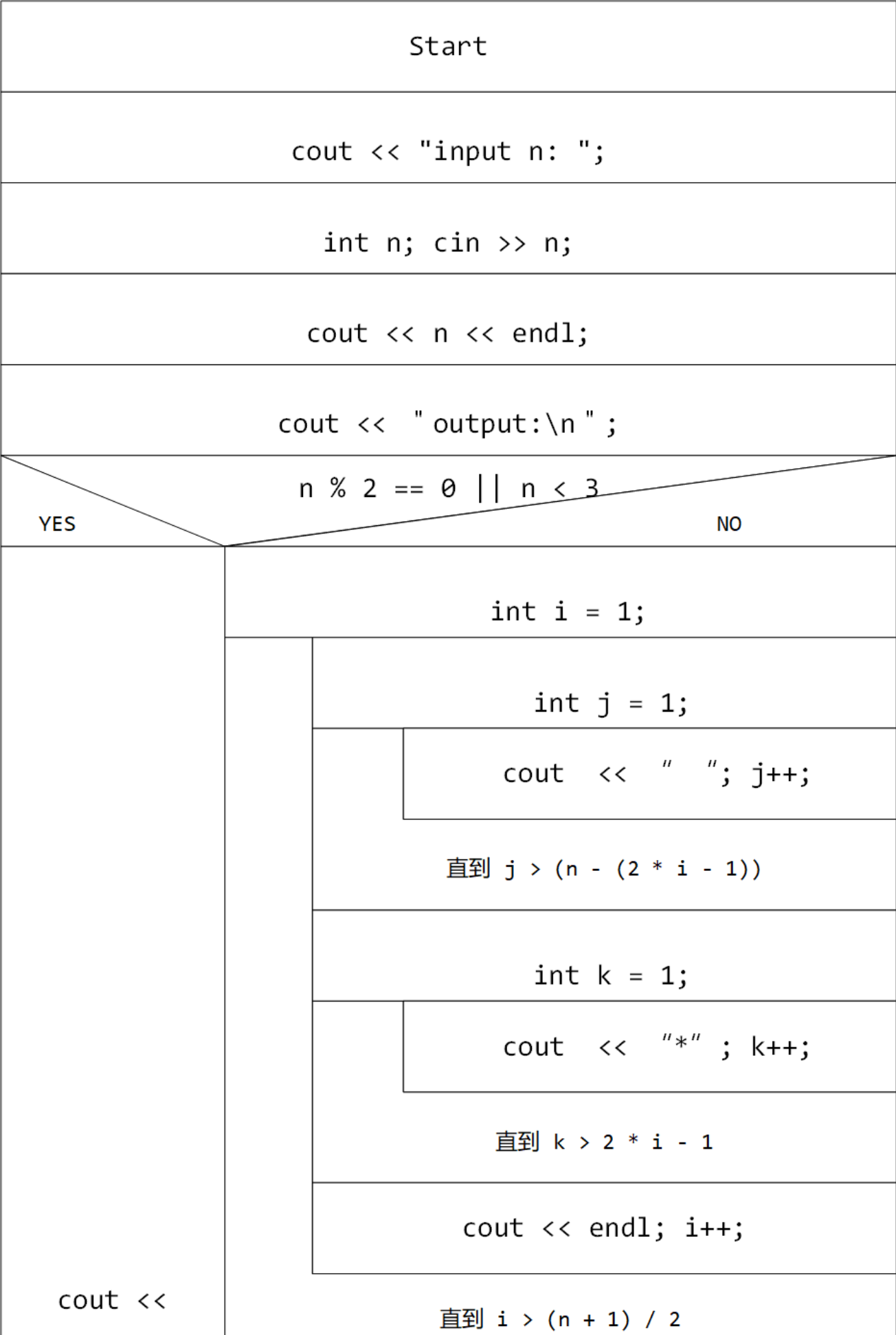
分析

输出的菱形从上到下为奇数列 $1, 3, 5, \dots, n-2, n, n-2, \dots, 3, 1$, 即共有 n 行, 第 i 行的`*`的数量满足函数:

$$N(i) = \begin{cases} 2i - 1 & 0 < i \leq \frac{n+1}{2} \\ 2(n+1-i) - 1 & \frac{n+1}{2} < i < n \end{cases}$$

在`*`前补充相应数量的(空格)即可得到菱形的形状

所以得到流程图:



"Input Error \n\n";		
	int i = (n + 1) / 2 + 1;	
		int t = n + 1 - i;
		int j = 1;
		cout << " "; j++;
		直到 j > (n - (2 * t - 1)) / 2
		int k = 1;
		cout << "*" ; k++;
		直到 k > 2 * t - 1
		cout << endl; i++;
		直到 i > n
	End	

代码

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6      cout << "input n: ";
7      cin >> n;
8      cout << n << endl;
9      cout << "output:\n";

```

```
10     if (n % 2 == 0 || n < 3) cout << "Input Error\n";
11     else {
12         for (int i = 1; i <= (n + 1) / 2; i++) {
13             for (int j = 1; j <= (n - (2 * i - 1)) / 2; j++) {
14                 cout << " ";
15             }
16             for (int k = 1; k <= 2 * i - 1; k++) {
17                 cout << "*";
18             }
19             cout << endl;
20         }
21         for (int i = (n + 1) / 2 + 1; i <= n; i++) {
22             int t = n + 1 - i;
23             for (int j = 1; j <= (n - (2 * t - 1)) / 2; j++) {
24                 cout << " ";
25             }
26             for (int k = 1; k <= 2 * t - 1; k++) {
27                 cout << "*";
28             }
29             cout << endl;
30         }
31     }
32     return 0;
33 }
```

结果


```
D:\Files...
input n: 1
1
output:
Input Error

input n: 2
2
output:
Input Error

input n: 3
3
output:
 *
***
 *

input n: 4
4
output:
Input Error

input n: 5
5
output:
 *
***
*****
***
 *

input n: 7
7
output:
 *
***
*****
*****
*****
***
 *
```

分析总结

只需找出各行 * 数量的关系便可以实现本程序

第三题：哥德巴赫猜想

实验目的

验证哥德巴赫猜想：对于任意一个大于或等于4的偶数 X ，至少存在一对素数 p_1 和 p_2 ，使得 $X = p_1 + p_2$ 。要求输入任意给定的整数 N ，若 N 是大于或者等于4的偶数，则输出一组分解情况 $N = p_1 + p_2$ 。为保证分解唯一性，规定 p_1 为满足分解要求的最小值。若 N 不是大于或等于4的偶数，则输出错误提示 `Input Error`。

- 输入：整数 N
- 输出：分解表达式 `N=p1+p2` 或错误提示 `Input Error`

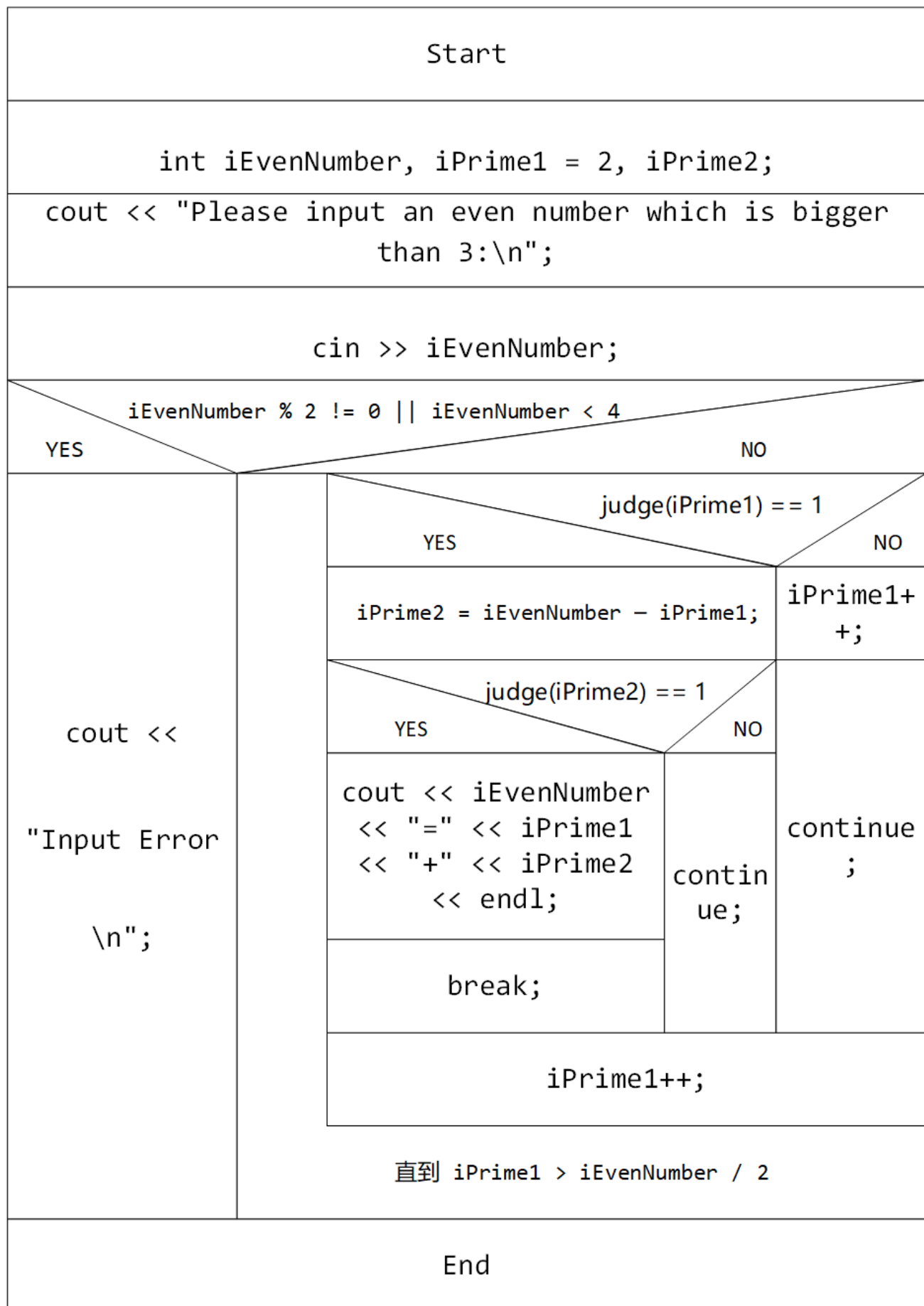
实验内容

分析

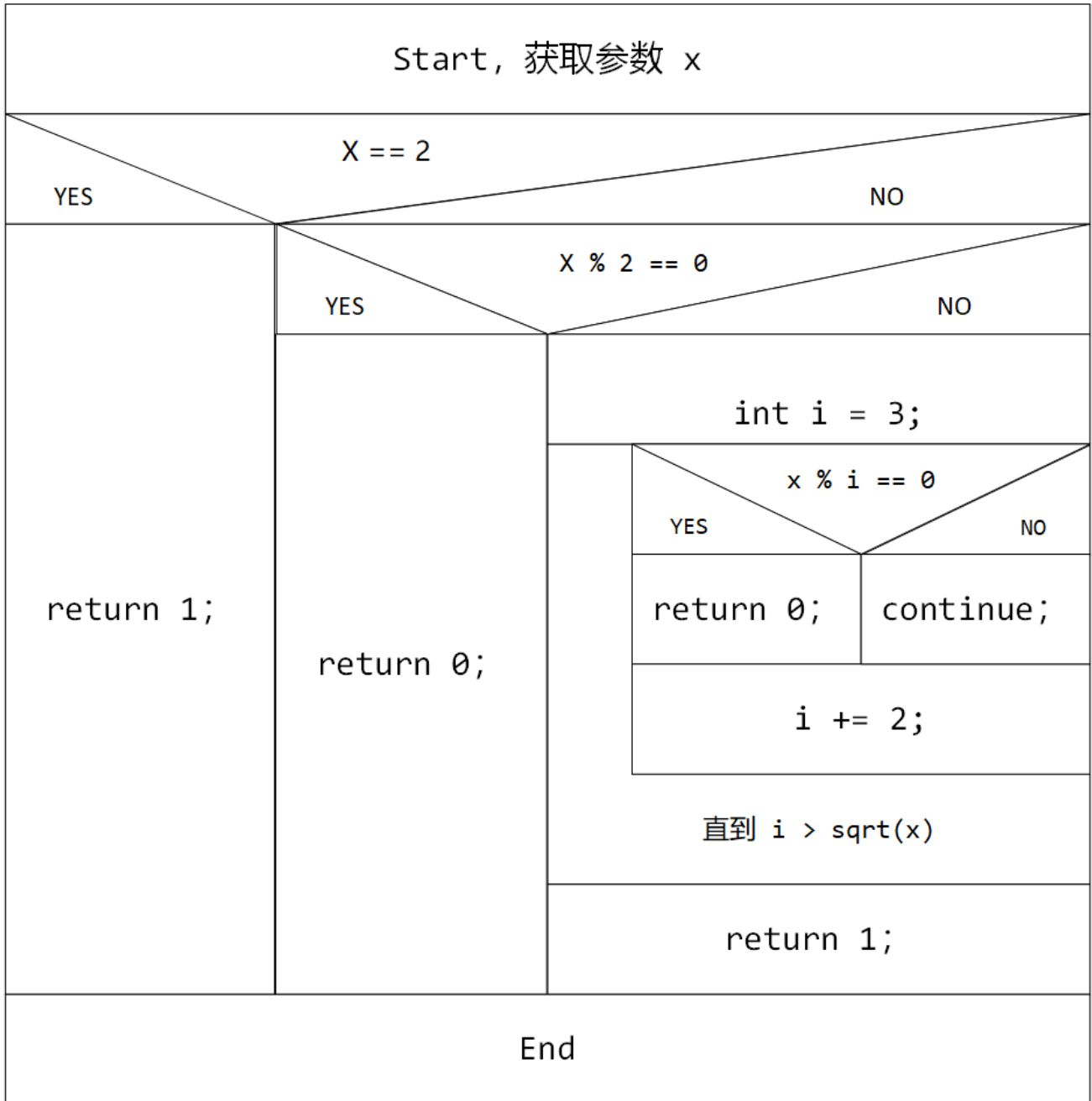
对每一个输入的偶数 N ，都将 p_1 从2开始遍历到 $\frac{N}{2}$ ，同时让 $p_2 = N - p_1$ （由题意 $p_1 \leq p_2$ ），再分别验证 p_1 和 p_2 是否为素数，如果是素数，输出即可。

而对于验证是否为素数，与**李生素数**一题有所不同，因为输入的 N 大小无限制，所以无法轻易给出一个足够使用的素数表来对 p_1 和 p_2 进行判断。故采用循环检验2和所有小于 p_1 和 p_2 的奇数是否为它们的因数的方法来给出判断结果。

流程图如下：



判断是否为素数的函数 `bool judge(int x)`



代码

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  bool judge(int x);
6
7  int main() {
8      int iEvenNumber, iPrime1 = 2, iPrime2;
9      cout << "Please input an even number which is bigger than 3:\n";
10     cin >> iEvenNumber;
11
12     if (iEvenNumber % 2 != 0 || iEvenNumber < 4)cout << "Input Error\n";
```

```

13     else {
14         for (iPrime1; iPrime1 <= iEvenNumber / 2; iPrime1++) {
15             if (judge(iPrime1) == 1) {
16                 iPrime2 = iEvenNumber - iPrime1;
17                 if (judge(iPrime2) == 1) {
18                     cout << iEvenNumber << "=" << iPrime1 << "+" << iPrime2 <<
endl;
19                     break;
20                 }
21             }
22         }
23     }
24
25     return 0;
26 }
27
28 bool judge(int x) {
29     if (x == 2) return 1;
30     else if (x % 2 == 0) return 0;
31     else {
32         for (int i = 3; i <= sqrt(x); i += 2) {
33             if (x % i == 0) return 0;
34         }
35         return 1;
36     }
37 }

```

结果

```
D:\Files\课程\程设\作业\homework\Debug\home...
Please input an even number which is bigger than 3:
1
Input Error
Please input an even number which is bigger than 3:
2
Input Error
Please input an even number which is bigger than 3:
3
Input Error
Please input an even number which is bigger than 3:
4
4=2+2
Please input an even number which is bigger than 3:
5
Input Error
Please input an even number which is bigger than 3:
6
6=3+3
Please input an even number which is bigger than 3:
8
8=3+5
Please input an even number which is bigger than 3:
10
10=3+7
Please input an even number which is bigger than 3:
98
98=19+79
Please input an even number which is bigger than 3:
102
102=19+83
```

分析总结

与**李生素数**一题相似，将判断是否为素数的部分单独写为一个函数可以简化主要函数构造的思考。

选做题1：分苹果

分析

将题给条件转换为代数式化简后如代码中所写

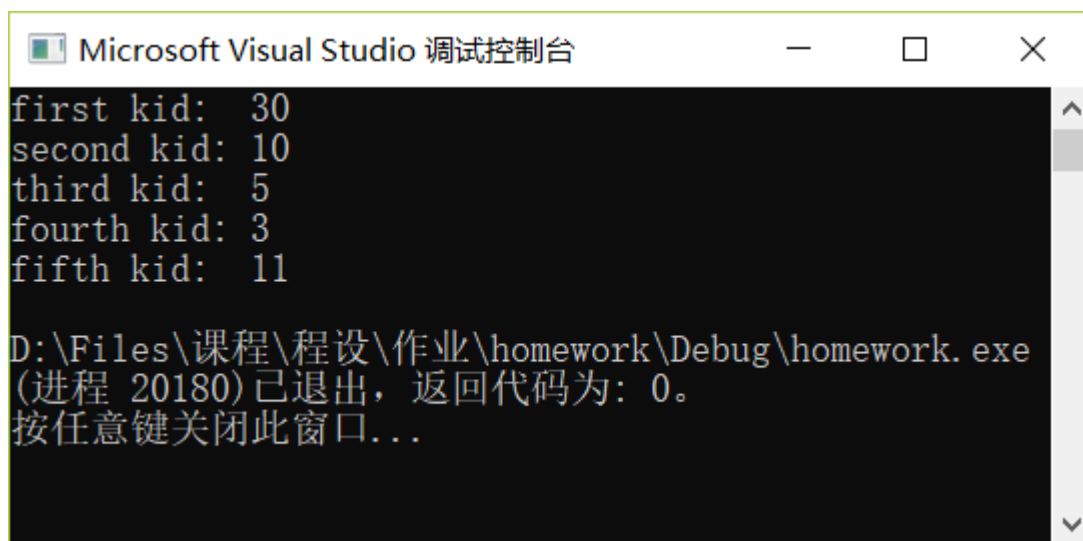
代码

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      double n = 11;
6      for (n;; n++) {
7          if (((n + 1) / 2 - (int)((n + 1) / 2)) == 0) {
8              if (((n + 1) / 6 - (int)((n + 1) / 6)) == 0) {
9                  if (((n + 1) / 12 - (int)((n + 1) / 12)) == 0) {
10                     if (((n + 1) / 20 - (int)((n + 1) / 20)) == 0) {
11                         if (((n - 4) / 5 - 11) == 0) {
12                             int t = n, fk = (n + 1) / 2, sk = (n + 1) / 6, tk = (n
+ 1) / 12, fok = (n + 1) / 20;
13                             cout << "total:      " << n << endl;
14                             cout << "first kid:  " << (n + 1) / 2 << endl;
15                             cout << "second kid: " << (n + 1) / 6 << endl;
16                             cout << "third kid:  " << (n + 1) / 12 << endl;
17                             cout << "fourth kid: " << (n + 1) / 20 << endl;
18                             cout << "fifth kid:  " << 11 << endl;
19                             break;
20                         }
21                     }
22                 }
23             }
24         }
25     }
26     return 0;
27 }

```

结果



```

Microsoft Visual Studio 调试控制台
first kid: 30
second kid: 10
third kid: 5
fourth kid: 3
fifth kid: 11

D:\Files\课程\程设\作业\homework\Debug\homework.exe
(进程 20180)已退出, 返回代码为: 0。
按任意键关闭此窗口...

```

选做题2：函数

分析

按照题目直接定义两个函数后在 `main` 中调用即可

代码

```
1  #include <iostream>
2  using namespace std;
3
4  int k(int n);
5  int S(int n);
6
7  int main() {
8      cout << "input n: ";
9      int n;
10     cin >> n;
11     cout << "s(n) = " << S(n) << "\n\n";
12
13     return 0;
14 }
15
16 int k(int n) {
17     int s = n;
18     for (int i = 2; i < n; i++) {
19         s *= i;
20     }
21     return s;
22 }
23
24 int S(int n) {
25     int s = 0;
26     for (int i = 1; i <= n; i++) {
27         s += k(i);
28     }
29     return s;
30 }
```

结果


```
S(n) = 1

input n: 2
S(n) = 3

input n: 3
S(n) = 9

input n: 4
S(n) = 33

input n: 5
S(n) = 153

input n: 6
S(n) = 873

input n: 7
S(n) = 5913

input n: 8
S(n) = 46233

input n: 9
S(n) = 409113

input n: 10
S(n) = 4037913

input n:
```