

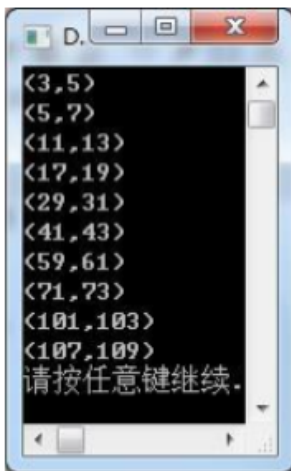
实验8

1 (10分)

编一判断m是否为素数的函数，并在主函数中利用它输出十对最小的孪生素数。所谓孪生素数是指两个相差为2的素数，如3和5，11和13。程序运行结果见下图。

函数形式为：

bool isprime(int m)



```
1 #include <iostream>
2 using namespace std;
3
4 // 判断是否为素数
5 bool isprime(int m) {
6     if (m <= 1) return false; // 1及以下的数字不是素数
7     for (int i = 2; i * i <= m; i++) { // 只检查到sqrt(m)即可
8         if (m % i == 0) {
9             return false; // 如果能被i整除，则m不是素数
10        }
11    }
12    return true; // 否则，m是素数
13 }
14
15 int main() {
16     int count = 0; // 计数器，用于输出10对孪生素数
17     int num = 2;   // 从2开始判断素数
18     while (count < 10) {
19         if (isprime(num) && isprime(num + 2)) { // 判断num和num+2是否是孪生素数
20             cout << "<" << num << "," << num + 2 << ">" << endl;
21             count++; // 输出一对后，计数器加1
22         }
23         num++;
24     }
25 }
```

```

22     }
23     num++; // 继续检查下一个数字
24 }
25
26 return 0;
27 }

```

解释：

1. `isprime(int m)`：这个函数用于判断数字 `m` 是否为素数。通过检查从2到 `sqrt(m)` 的所有整数，如果 `m` 能被其中某个数整除，则说明 `m` 不是素数。
2. 主函数 `main()`：我们使用一个 `while` 循环，检查连续的整数 `num` 和 `num + 2` 是否是孪生素数。如果它们都是素数，就输出这一对孪生素数，并继续寻找下一对，直到输出 10 对为止。

D:\2024\课内\C++\第八章\8.1\Debug\8.1.exe

```

<3, 5>
<5, 7>
<11, 13>
<17, 19>
<29, 31>
<41, 43>
<59, 61>
<71, 73>
<101, 103>
<107, 109>
请按任意键继续. . .

```

2 (10分)

编一函数，功能为判断一字符串是否为回文，如果是回文则返回1，否则返回0。回文是指顺读和倒读都一样的字符串，如“deed”和“level”是回文。在主函数中对输入的字符串加以调用。

函数形式为：int huiwen(char s[])

```

1 #include <iostream>
2 #include <cstring> // 用于strlen函数
3 using namespace std;
4
5 // 判断字符串是否为回文
6 int huiwen(char s[]) {
7     int len = strlen(s); // 获取字符串的长度
8     for (int i = 0; i < len / 2; i++) {
9         if (s[i] != s[len - 1 - i]) { // 如果对应位置的字符不相同
10             return 0; // 不是回文，返回0

```

```

11     }
12 }
13 return 1; // 是回文, 返回1
14 }
15
16 int main() {
17     char str[100]; // 定义一个字符串数组
18     cout << "请输入一个字符串: ";
19     cin >> str; // 输入字符串
20
21     // 调用 huiwen 函数判断字符串是否为回文
22     if (huiwen(str) == 1) {
23         cout << "该字符串是回文。" << endl;
24     } else {
25         cout << "该字符串不是回文。" << endl;
26     }
27     system("pause");
28     return 0;
29 }

```

解释:

1. **huiwen(char s[])**: 该函数接收一个字符串作为参数, 并通过逐个比较字符串的字符 (从首尾逐渐向中间) 来判断是否为回文。如果存在任何一对字符不相等, 函数返回0表示不是回文, 否则返回1表示是回文。
2. **strlen(s)**: 这个函数返回字符串的长度, 用于确定循环的范围。我们只需要检查字符串的前半部分和后半部分的字符是否相等。
3. **主函数**: 从用户那里输入一个字符串, 调用 **huiwen** 函数判断字符串是否为回文, 并输出结果。

D:\2024\课内\C++\第八章\8.2\Debug\8.2.exe

```

请输入一个字符串: 656
该字符串是回文。
请按任意键继续. . .

```

3 (20分)

函数的功能是将学生成绩从高分到低分排序，并统计优秀与不及格的人数。用下面两种方法实现：

(1) 函数形式为：int fun1(int s[],int n,int *x)

要求优秀人数通过return返回，不及格人数通过指针参数返回结果。

(2) 函数形式为：void fun2(int s[],int n,int &x,int &y)

要求优秀与不及格的人数通过引用参数返回结果。

分别编二个函数，学生人数从键盘输入。

方法1：通过返回值返回优秀人数，通过指针参数返回不及格人数

```
1  #include <iostream>#include <algorithm> // 用于 sort 函数using namespace std;
2
3  // (1) 通过返回值返回优秀人数，指针返回不及格人数int fun1(int s[], int n, intx)
   {int excellent = 0, fail = 0;
4      // 排序，按照从高到低sort(s, s + n, greater<int>());
5      // 统计优秀与不及格的人数for (int i = 0; i < n; i++) {if (s[i] >= 90) {
6          excellent++; // 优秀
7          }if (s[i] < 60) {
8              fail++; // 不及格
9          }
10     }
11     *x = fail; // 通过指针返回不及格人数return excellent; // 返回优秀人数
12 }
13
14 int main() {int n;
15     cout << "请输入学生人数：";
16     cin >> n;
17     ints = new int[n]; // 动态分配一个数组来存储学生成绩
18     cout << "请输入每个学生的成绩：";for (int i = 0; i < n; i++) {
19         cin >> s[i];
20     }
21     // 方法1：使用返回值和指针来获取统计数据int fail1;int excellent1 = fun1(s, n,
        &fail1);
22     cout << "优秀人数 = " << excellent1 << ", 不及格人数 = " << fail1 << endl;
23     deletes; // 释放动态分配的内存system("pause");return 0;
24 }
```

关键点说明：

- `fun1` 通过返回值返回优秀人数 (≥ 90)，通过指针返回不及格人数 (< 60)。
- 排序是通过 `sort(s, s + n, greater<int>())` 实现的，将成绩从高到低排序。
- 动态内存分配 `new int[n]` 和释放 `delete[] s`。

方法2：通过引用参数返回优秀与不及格的人数

```
1 #include <iostream>#include <algorithm> // 用于 sort 函数using namespace std;
2
3 // (2) 通过引用参数返回优秀人数与不及格人数void fun2(int s[], int n, int &x, int
  &y) {
4     x = 0; // 优秀人数初始化为0
5     y = 0; // 不及格人数初始化为0// 排序，按照从高到低sort(s, s + n, greater<int>
  ());
6     // 统计优秀与不及格的人数for (int i = 0; i < n; i++) {if (s[i] >= 90) {
7         x++; // 优秀
8     }if (s[i] < 60) {
9         y++; // 不及格
10    }
11 }
12 }
13
14 int main() {int n;
15     cout << "请输入学生人数: ";
16     cin >> n;
17     ints = new int[n]; // 动态分配一个数组来存储学生成绩
18     cout << "请输入每个学生的成绩: ";for (int i = 0; i < n; i++) {
19         cin >> s[i];
20     }
21 // 方法2: 使用引用参数来获取统计数据int excellent2, fail2;fun2(s, n, excellent2,
  fail2);
22     cout << "优秀人数 = " << excellent2 << ", 不及格人数 = " << fail2 << endl;
23     deletes; // 释放动态分配的内存system("pause");return 0;
24 }
```

关键点说明：

- `fun2` 使用引用参数返回优秀人数和不及格人数。
- 使用 `sort(s, s + n, greater<int>())` 排序学生成绩。
- 通过引用参数 `x` 和 `y` 返回优秀和不及格的人数。

