

课前思考题

程序输出是什么？

```
1  #include <iostream>
2  using namespace std;
3
4  int* print(int* output, int data)
5  {
6      *output = data;
7      return ++output;
8  }
9
10 int f1()
11 {
12     cout << "f1";
13     return 1;
14 }
15
16 int f2()
17 {
18     cout << "f2";
19     return 2;
20 }
21
22 int f3()
23 {
24     cout << "f3";
25     return 3;
26 }
27
```

```
28 int main()
29 {
30     /// CASE 1
31     int i = 0;
32     cout << i << i++ << i++;
33     cout << endl;
34
35     /// CASE 2
36     cout << f1() << f2() << f3();
37     cout << endl;
38
39     /// CASE 3
40     int output[10];
41     print(print(print(output, f1()), f2()), f3());
42     cout << endl;
43
44     for (int i=0; i<3; i++)
45         cout << output[i] << ' ';
46     cout << endl;
47
48     return 0;
49 }
```

```
D:\>g++ example-20121024.cpp
```

```
D:\>a
```

```
210
```

```
f3f2f1123
```

```
f3f2f1
```

```
1 2 3
```

```
D:\> 你做对了吗？
```

思考题给我们的启示

- 在程序设计与调试中出现的令人困惑的各种诡异现象，实际上都是有原因的。
- 对于新知识，要掌握设计实现专门的短小程序来验证的方法，加深理解。
- 加强对程序输出结果进行合理“解释”的能力，必要时可改造程序进行验证。

6.2 筛法

- 筛法的算法思想与依据
- 筛法的程序实现特点
- 筛法与枚举思想的区别与联系
- 做标记、打标签的编程技术

6.2 筛法

任务二、求100以内的所有素数

什么是“筛法”？

1. 想象将100个数看作沙子和小石头子，让小石头子称素数；让沙子当作非素数。弄一个筛子，只要将沙子筛走，剩下的就是素数了。
2. 合数一定是 2、3、4 的倍数。
3. 使用数组，让下标就是100以内的数，**让数组元素的值作为筛去与否的标志**。比如筛去以后让元素值为1。

		0	0	1	0	1	0	...	1	1
0	1	2	3	4	5	6	7		99	100

方法的依据：

1至100这些自然数可以分为三类：

- 单位数：仅有一个数1。
- 素数： 大于1，且只有1和它自身这样两个正因数。
- 合数： 除了1和自身以外，还有其他正因数。

1不是素数，除1以外的自然数，当然只有素数与合数。筛法实际上是筛去合数，留下素数。

为了提高筛法效率，根据初等数论知识，如果 n 是合数，则必定存在如下关系式：

$$1 < c \leq \sqrt{n}$$

c 是合数 n 的最小正因数。

因此，只要是 c 的倍数的数，一定是合数，应该将其筛去。

一定注意：要进行“筛”的1—100的数字是与数组`prime[101]`的下标相对应的，而每个数组元素的取值只有2个：是0或1，分别代表（标志）与下标相对应的数字是素数或不是素数。

筛合数的过程



0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

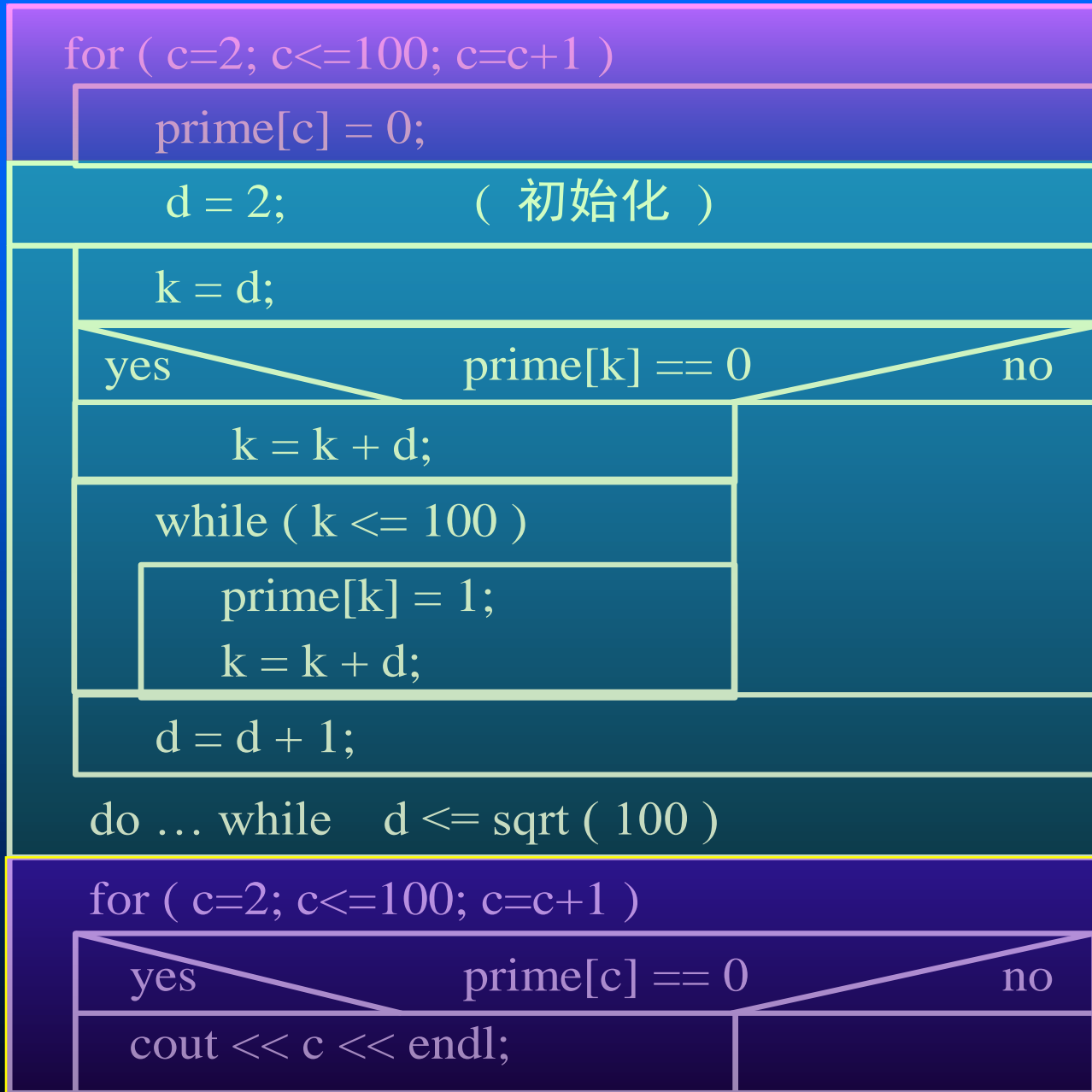
0	0	1	0	1	0	1	1	1	0	1	0	1	1	1
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

寻找2的倍数

寻找3的倍数

寻找4的倍数（跳过）

程序框图如下：



请分析左
边程序的
结构

从而了解
算法的设
计思路

为程序代
码的实现
创造条件

上述框图很清晰地描述了筛法的思路：

- STEP 1. 将prime数组清零，使用了一个计数型的循环语句。
- STEP 2. 将正因数d初始化为 $d = 2$ 。
- STEP 3. 循环筛数。这里用了一个 do while 语句，也称为“直到型循环”。
- STEP 4. 输出“筛”过后的数组，即剩下的所有素数，这里使用了标识数组。

补充说明

- 在do...while语句中，检测prime[k]是否为0。若是0，则说明k是质数。原因如下：
 - 显然， $\text{sqrt}(k) < k$ ，故当检测到k时， $2 \sim \text{sqrt}(k)$ 中所有的数必然已全部被测试过了。
 - 在 $2 \sim \text{sqrt}(k)$ 区间中，要么有整除k的数，要么没有。
 - 若有整除k 的数，则k是该数的倍数，prime[k]必会被设为 1 。反之，若prime[k]没有被设成1，则说明区间中必没有整除k的数，这样，根据初等数论，k必是质数。

根据程序流程图编写源程序

// 输出1—100中所有的质数

// 使用“筛”法思想来实现

```
#include <iostream> // cout
```

```
#include <cmath>     // sqrt()
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int prime[101] = {1, 1}; // why 101 ?
```

```
    // 0,1肯定不是质数，先筛去，相应位置的标记设为1.
```

```
int d = 2; // 从第一个质数2开始，筛除它们的倍数
do {
    int k = d;
    if (prime[k] == 0) { // 只使用没被筛掉的质数
        k += d;          // k 是 d 的倍数
        while (k <= 100) {
            prime[k] = 1; // 被“筛”掉了
            k += d;       // k = k + d;
        }
    }
    d++; // 下一个 d
} while (d <= sqrt(100));
```

如何将do..while, while循环都改成for循环?

```
for (int c=2; c<=100; c++)  
{  
    if (prime[c] == 0)  
        cout << c << endl; //why not prime[c] ?  
}
```

```
return 0;  
}
```

如何

将do..while, while循环都改成for循环

课后阅读


```
for (int d=2; d<=sqrt(100); d=d+1)
{
    int k = d;
    if (prime[k] == 0) {
        for (k=k+d; k<=100; k=k+d)
            prime[k] = 1; // 被“筛”掉了
    }
}
```

简化

```
int last = sqrt(100);
for (int d=2; d<=last; d++)
{
    if (prime[d] == 0) {
        for (int k=d*2; k<=100; k+=d)
            prime[k] = 1;
    } // IF
} // FOR
```

```
for (int d=2; d<=sqrt(100); d=d+1)
{
    int k = d;
    if (prime[k] == 0) {
        for (k=k+d; k<=100; k=k+d)
            prime[k] = 1; // 被“筛”掉了
    }
}
```

简化

```
int last = sqrt(100);
for (int d=2; d<=last; d++)
{
    if (prime[d] == 1) continue;
    for (int k=d*2; k<=100; k+=d)
        prime[k] = 1;
} // FOR
```

```
for (int d=2; d<=sqrt(100); d=d+1)
{
    int k = d;
    if (prime[k] == 0) {
        for (k=k+d; k<=100; k=k+d)
            prime[k] = 1; // 被“筛”掉了
    }
}
```

继续简化

```
int last = sqrt(100);
for (int d=2; d<=last; d++)
{
    if (prime[d]) continue;
    for (int k=d*2; k<=100; k+=d)
        prime[k] = 1;
} // FOR
```

结 束