目标：

1. 编写C++程序来达到：输入整数x和y，通过规则进行计算得到遗传结果r。
2. 编写C++程序来达到：输入正整数，输出其间的质数（包括这两个数本身）。

原理：

1. 通过位运算进行操作。同时在操作的过程中可以用与2的n次方进行按位与操作来输出除第n-1位均为0，第n-1位不变的整数。这样可以将按位遍历运算的操作化为循环计算求和。
2. 用vector来实现对结果的存储和自动测试功能，在判断一个数是否是质数时使用小于该数且大于1的所有数进行取余操作。

**如何编程和运行：**

* 软件构件库部分：

构建所需的类和最基本的计算函数。

第一题中使用了位运算操作，同时用2的幂次进行按位与操作来得到某一位的数据。

第二题中使用vector，同时遍历从2到（x - 1）的所有数来达到判断一个数是否是质数的效果。

* 构件库测试部分：

编写所需要的测试函数，自动测试部分还包括测试所需要的一些数据。

第一题中的自动测试程序提供了输出操作数的二进制表示的功能，第二题的自动测试程序还包括大质数的检测部分。

* 构件库应用部分：

编写main函数，在函数中提供选择自动和手动测试的功能。

结果：

以下测试数据均在64位操作系统下用VS 2017 测试得到。

第一题自动测试程序运行结果：

测试开始！

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

现在测试第 1 组数据， 351 和 15 的结果计算为 -969831300 ，计算正确！详细结果如下：

00000000000000000000000101011111

00000000000000000000000000001111

11000110001100011000110001111100

现在测试第 2 组数据， 68415 和 185 的结果计算为 -969830706 ，计算正确！详细结果如下：

00000000000000010000101100111111

00000000000000000000000010111001

11000110001100011000111011001110

现在测试第 3 组数据， -952 和 -69 的结果计算为 415642361 ，计算正确！详细结果如下：

11111111111111111111110001001000

11111111111111111111111110111011

00011000110001100011001011111001

现在测试第 4 组数据， -486 和 15345 的结果计算为 -1385467741 ，计算正确！详细结果如下：

11111111111111111111111000011010

00000000000000000011101111110001

10101101011010110111000010100011

现在测试第 5 组数据， 0 和 99999 的结果计算为 -969896203 ，计算正确！详细结果如下：

00000000000000000000000000000000

00000000000000011000011010011111

11000110001100001000111011110101

现在测试第 6 组数据， 68 和 2147483647 的结果计算为 -346368331 ，计算正确！详细结果如下：

00000000000000000000000001000100

01111111111111111111111111111111

11101011010110101101011010110101

现在测试第 7 组数据， 564 和 -2147483648 的结果计算为 1177652823 ，计算正确！详细结果如下：

00000000000000000000001000110100

10000000000000000000000000000000

01000110001100011000111001010111

现在测试第 8 组数据， 68 和 25 的结果计算为 -969831305 ，计算正确！详细结果如下：

00000000000000000000000001000100

00000000000000000000000000011001

11000110001100011000110001110111

现在测试第 9 组数据， 15 和 35 的结果计算为 -969831324 ，计算正确！详细结果如下：

00000000000000000000000000001111

00000000000000000000000000100011

11000110001100011000110001100100

现在测试第 10 组数据， 94 和 68 的结果计算为 -969831369 ，计算正确！详细结果如下：

00000000000000000000000001011110

00000000000000000000000001000100

11000110001100011000110000110111

现在测试第 11 组数据， 6666 和 99 的结果计算为 -969826783 ，计算正确！详细结果如下：

00000000000000000001101000001010

00000000000000000000000001100011

11000110001100011001111000100001

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

共测试11组数据，其中0组错误！

第一题自动测试程序运行结果：

测试开始！先测试7559以内的质数！

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

现在测试第 1 组数据， 1 和 15 的结果。正确！

现在测试第 2 组数据， 5185 和 185 的结果。正确！

现在测试第 3 组数据， 1358 和 3333 的结果。正确！

现在测试第 4 组数据， 9 和 5345 的结果。正确！

现在测试第 5 组数据， 16 和 6999 的结果。正确！

现在测试第 6 组数据， 42 和 3512 的结果。正确！

现在测试第 7 组数据， 15 和 2530 的结果。正确！

现在测试第 8 组数据， 68 和 45 的结果。正确！

现在测试第 9 组数据， 15 和 35 的结果。正确！

现在测试第 10 组数据， 94 和 2 的结果。正确！

现在测试第 11 组数据， 7559 和 3 的结果。正确！

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

共测试11组数据，其中0组错误！

接下来测试大质数！

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

现在测试第 1 组数据， 9987843 和 9988607 的结果。正确！

现在测试第 2 组数据， 9989159 和 9997597 的结果。正确！

现在测试第 3 组数据， 9987840 和 9999289 的结果。正确！

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

共测试3组数据，其中0组错误！

自动测试程序验证结果报告：

* 对于64位的计算机

第一题：在测试的过程中发现对于给定样例没有错误的计算结果。

第二题：在用质数表和计算结果比对之后发现对于给定样例没有错误的计算结果。

对于手动输入程序，输出暂未发现问题，若输入不符合规则将要求重新输入。

结论：

这两个C++程序使用的自动测试程序暂未发现有运算错误的情况。