Psuedocode

```
Input(InputNumber)

For i = 1 to 128:

Lockers[]

factors = 0

For y = 1 to InputNumber

IF (i % y == 0)

factors = factors+1

ENDIF

EndFor

IF (factors % 2 == 0)

Lockers-append("X")

ELSE

Lockers-append("O")

ENDIF

EndFor
```

Testing

1. **128**

```
Student Number (1-128): 128
                                  When 128 is inputed, only the square numbers between 1
                                  and 128 inclusive should be opened (according to the
                                  math of this problem) That is;
                                  1
                                  4
                                  9
                                   16
                                  25
                                  36
                                  49
                                  64
                                  81
                                  100
                                  121
                                  The output shows all these lockers are opened
                                  accordingly.
```

2. 1 & 2

Following the logic of the riddle, all lockers should be open on 1 (since the first student opens them all), with every second locker being toggled closed on the second student's turn.

```
      Student Number (1-128): 1
      Student Number (1-128): 2

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0

      0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0
```

3. 3 & 4

When 3 is inputted, every third locker should be toggled. We can see that this does work in the program by comparing the "2" output to the "3." The same goes for 4, etc.

```
      Student Number (1-128): 3

      0 X X X 0 0 0 0 X

      X X 0 0 0 X X X 0

      0 0 0 X X X 0 0

      0 X X 0 0 0 X X 0 X

      0 X X 0 0 0 0 X X 0 X

      0 X X 0 0 0 X X X 0

      0 X X 0 0 0 X X X 0

      0 X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0

      0 X X X 0 0 0 X X X 0
```

4. 69, 70, 71

To be completely sure, we can test the program with random numbers in between 1 and 128. If we chose 69, 70, and 71, we should see that only the 70th locker changes for 70, and only the 71st locker changes for 71, etc.

```
Student Number (1-128): 69
                          Student Number (1-128): 70
                                                    Student Number (1-128): 71
                                    0 0
 0 0 0
                                                     00000
                           0000
 000
          0 0
                           0000
 0000
          0 0
                           00000
                                                     000000
 0 0 X 0
          0 0
          b o
 00000
                           000000
                                                     0000000
```

5. Extras

The program only reads valid data (an integer) and only functions if the integer is between 1 and 128 inclusive. It'll give an error message if a non-integer is entered, a message if an integer outside the domain is entered, and it will close when "0" is entered.

```
Student Number (1-128): -30

Please enter a number between 1 and 128 inclusive or 0 to end

Student Number (1-128): 80.4

Please enter a number between 1 and 128 inclusive or 0 to end

Student Number (1-128): 129

Invalid Number

Student Number (1-128): str

Please enter a number between 1 and 128 inclusive or 0 to end

Student Number (1-128): 0

cyas
```