

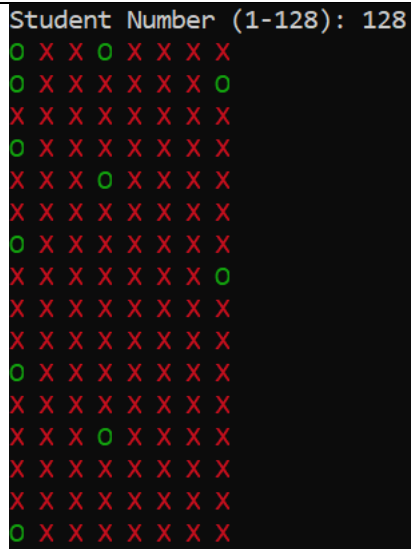
Psuedocode

```
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



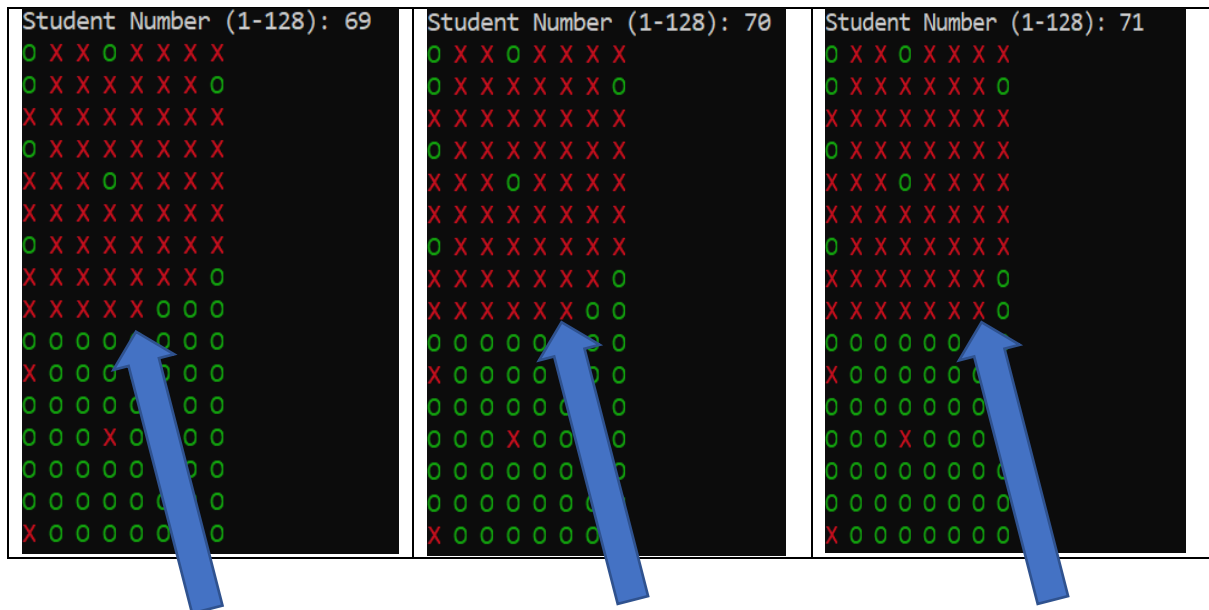
When 128 is inputted, only the square numbers between 1 and 128 inclusive should be opened (according to the math of this problem) That is;

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.



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
```