**Exercise #1**

Convert to reverse binary

Write a program that takes in a positive integer as input, and outputs a string of 1's and 0's representing the integer in reverse binary. For an integer x, the algorithm is:

As long as x is greater than 0

Output x modulo 2 (the remainder is either 0 or 1)

Assign x with x divided by 2

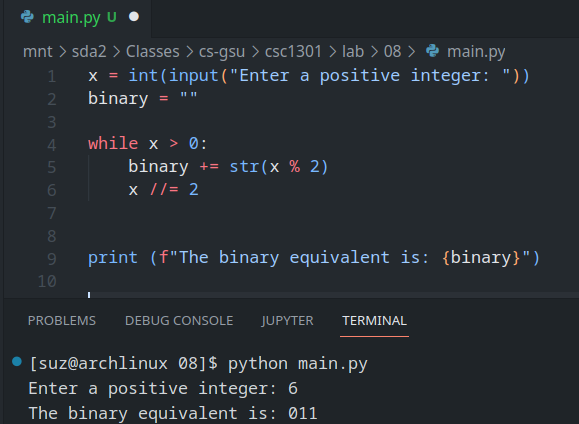
Note: The above algorithm outputs the 0's and 1's in reverse order.

Ex: If the input is: 6

the output is: 011

6 in binary is 110; the algorithm outputs the bits in reverse.

main.py



**Exercise #2**

Password modifier

Many user-created passwords are simple and easy to guess. Write a program that takes a simple password, and makes it stronger by replacing characters using the key below, and by appending "!" to the end of the input string.

i becomes 1

a becomes @

m becomes M

B becomes 8

s becomes $

Ex: If the input is:

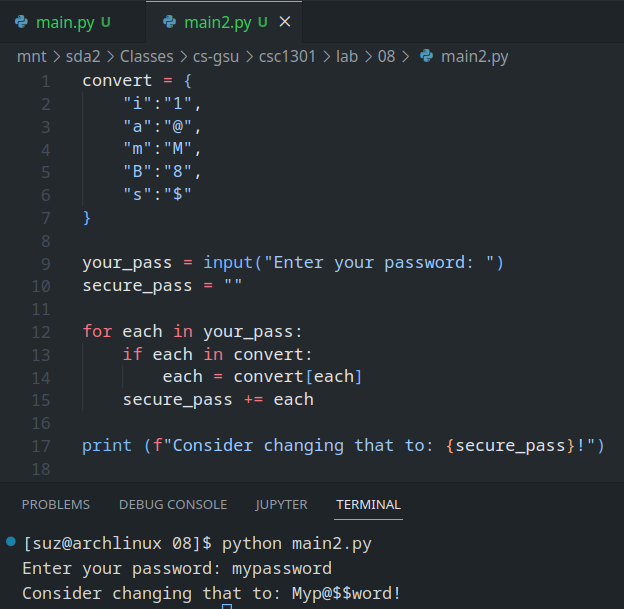
mypassword

the output is:

Myp@$$word!

Hint: Python strings are immutable, but support string concatenation. Store and build a stronger password in the given password variable.

main.py



**Exercise #3**

Brute force equation solver

Numerous engineering and scientific applications require finding solutions to a set of equations. Ex: 8x + 7y = 38 and 3x - 5y = -1 have a solution x = 3, y = 2. Given integer coefficients of two linear equations with variables x and y, use brute force to find an integer solution for x and y in the range -10 to 10.

Ex: If the input is:

8

7

38

3

-5

-1

Then the output is:

x = 3 , y = 2

Use this brute force approach:

For every value of x from -10 to 10

For every value of y from -10 to 10

Check if the current x and y satisfy both equations. If so, output the solution, and finish.

Ex: If no solution is found, output:

There is no solution

Assume the two input equations have no more than one solution.

Note: Elegant mathematical techniques exist to solve such linear equations. However, for other kinds of equations or situations, brute force can be handy.

main.py

