Lab 6P: Value-Returning Functions and Modules_ 6P_VRFMX

Programming Exercises: GADDIS PYTHON 2e Text, Pages 235 – 237

Using the IDLE editor (FILE/New Window)……

Name: _____Jeff Couch__________

EXTRA CREDIT = 8 (220 Pts)

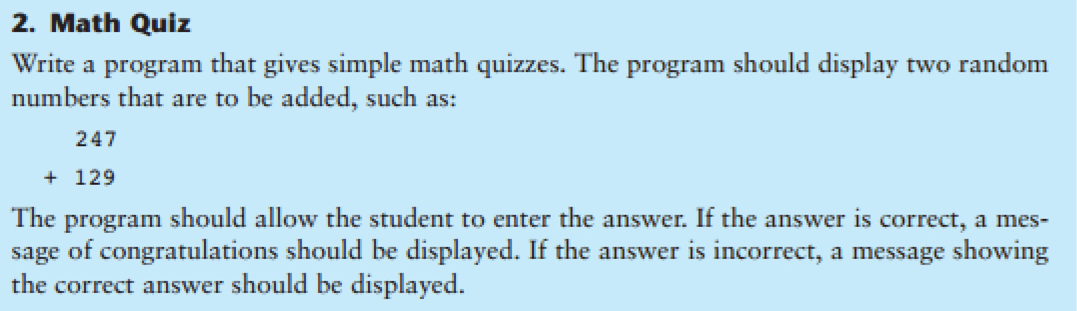
For EACH Project you complete:

1. Create a FLOWCHART for each program and attach a Screenshot(s) of your FLOWCHART.

2. Take one Screenshot of both the EDIT window disiplaying your program and the SHELL window displaying your program results and paste as directed.

SAVE your programs. All extra credit programs must be submitted with this lab template

Project 2. Math Quiz (20 Pts)

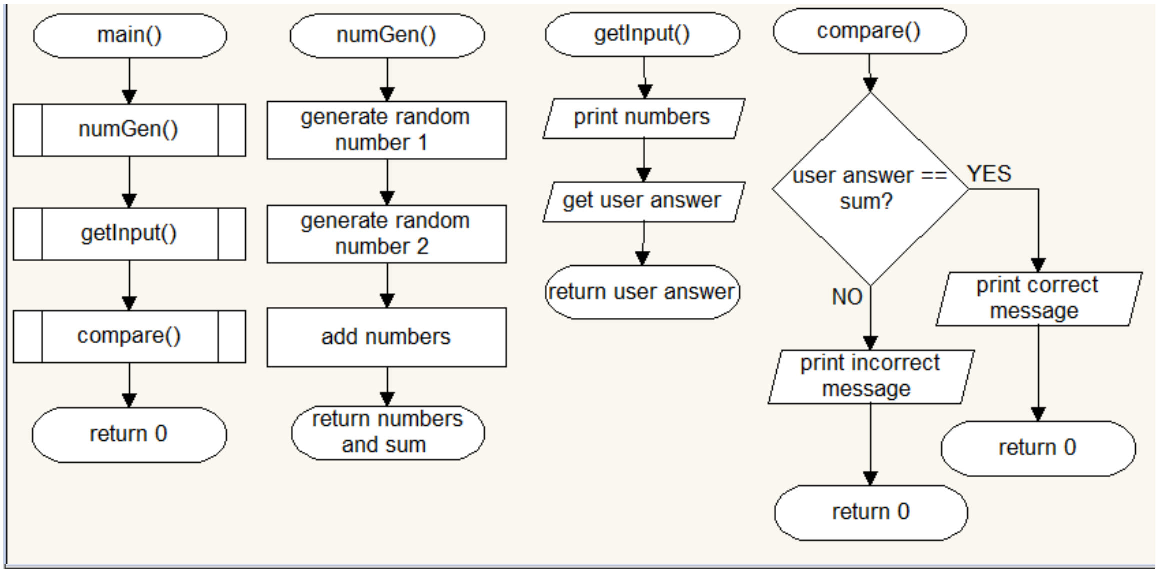


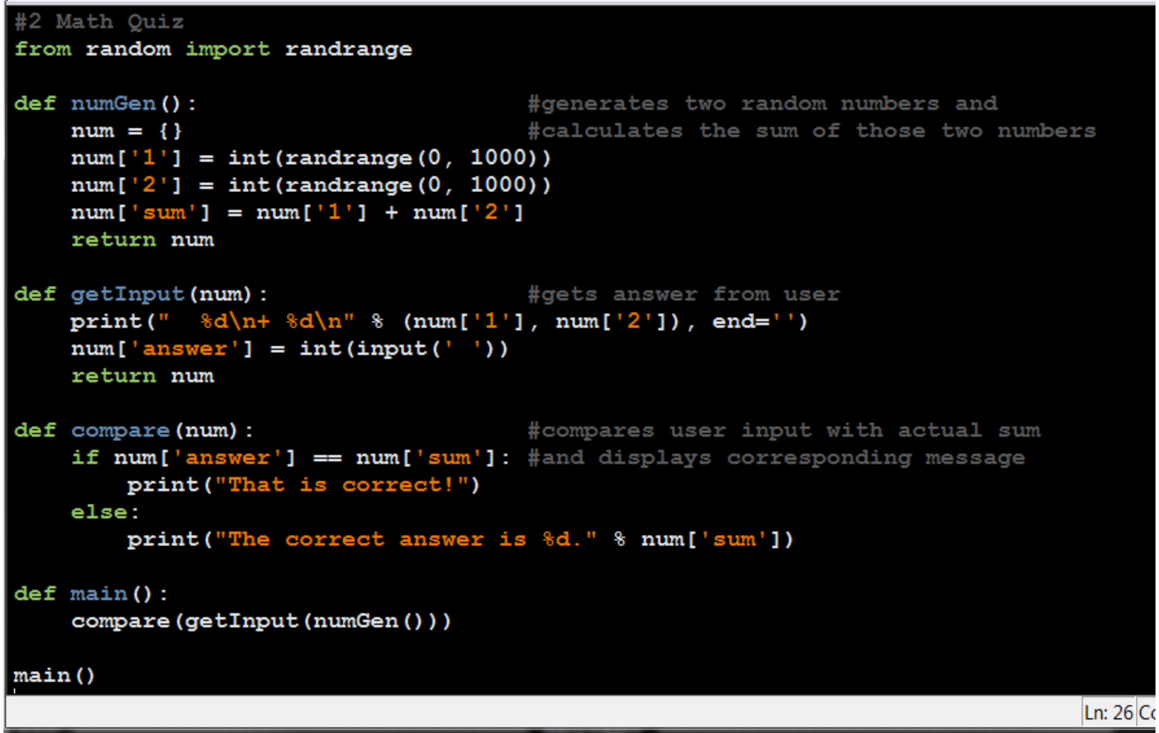
2. Math Quiz

Write a program that gives simple math quizzes. The program should display two random numbers that are to be added, such as:

```
     247
  +  129
```

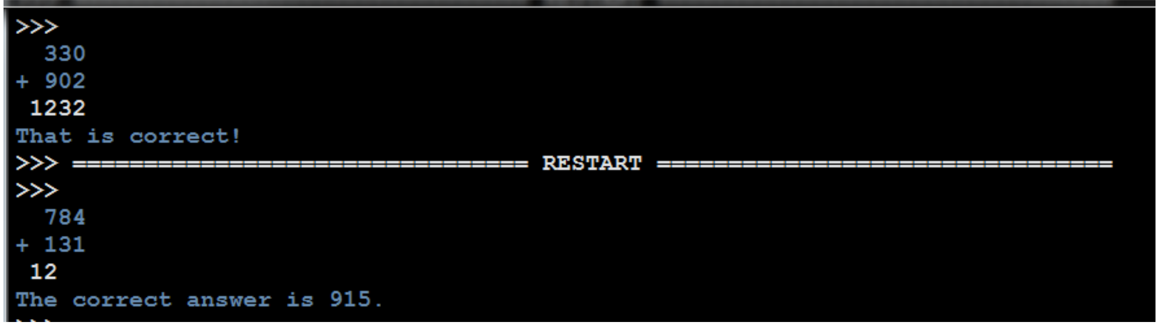
The program should allow the student to enter the answer. If the answer is correct, a message of congratulations should be displayed. If the answer is incorrect, a message showing the correct answer should be displayed.





```
#2 Math Quiz
from random import randrange

def numGen():                       #generates two random numbers and
    num = {}                        #calculates the sum of those two numbers
    num['1'] = int(randrange(0, 1000))
    num['2'] = int(randrange(0, 1000))
    num['sum'] = num['1'] + num['2']
    return num

def getInput(num):                  #gets answer from user
    print("  %d\n+ %d\n" % (num['1'], num['2']), end='')
    num['answer'] = int(input(' '))
    return num

def compare(num):                   #compares user input with actual sum
    if num['answer'] == num['sum']: #and displays corresponding message
        print("That is correct!")
    else:
        print("The correct answer is %d." % num['sum'])

def main():
    compare(getInput(numGen()))

main()
```

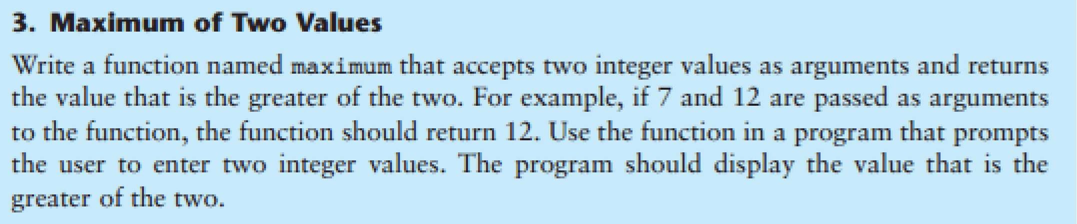
Ln: 26 Co



```
>>>
  330
+ 902
 1232
That is correct!
>>> ================================ RESTART ================================
>>>
  784
+ 131
 12
The correct answer is 915.
>>>
```

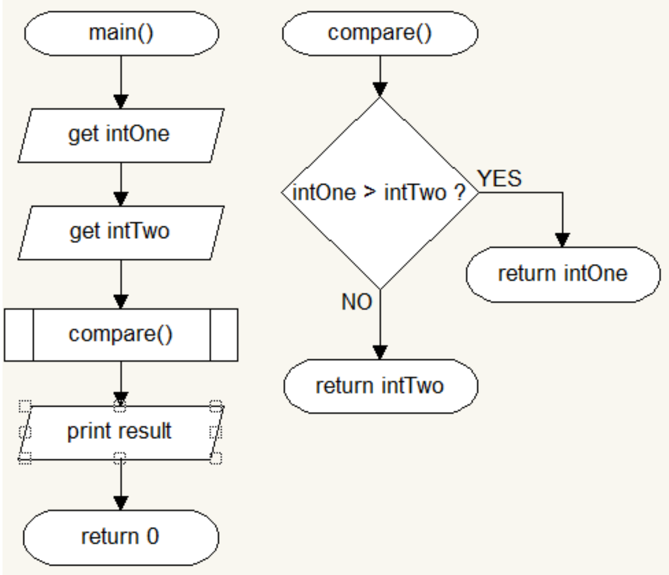
Project 3: Maximum of Two Values (20 Pts)

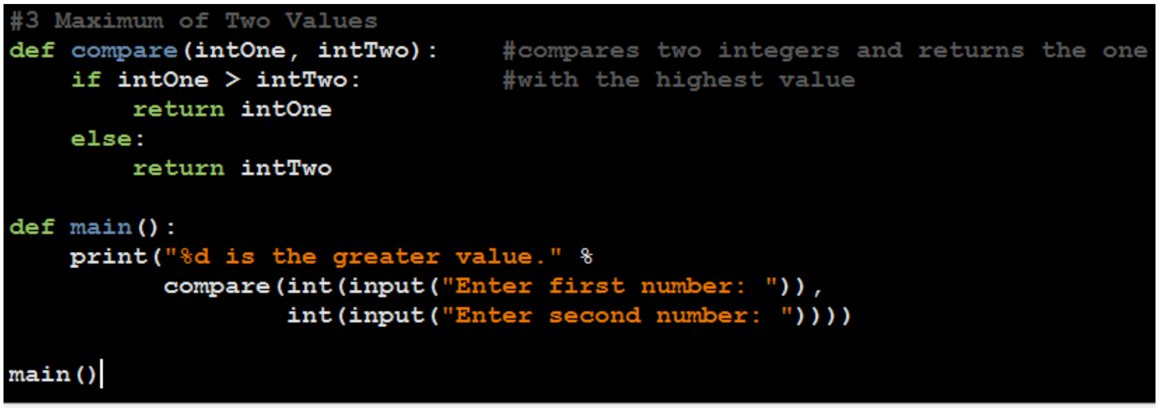
Input: Values = 89,99



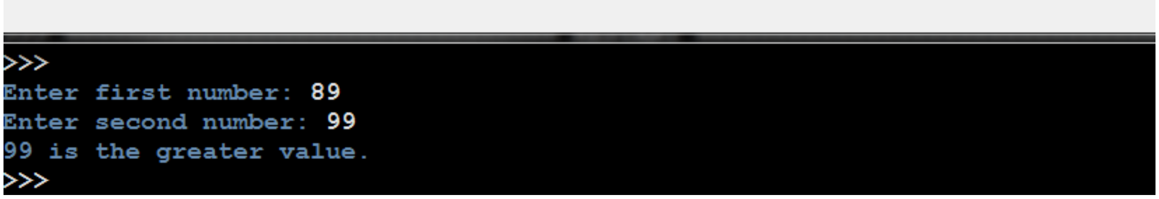
3. Maximum of Two Values

Write a function named `maximum` that accepts two integer values as arguments and returns the value that is the greater of the two. For example, if 7 and 12 are passed as arguments to the function, the function should return 12. Use the function in a program that prompts the user to enter two integer values. The program should display the value that is the greater of the two.



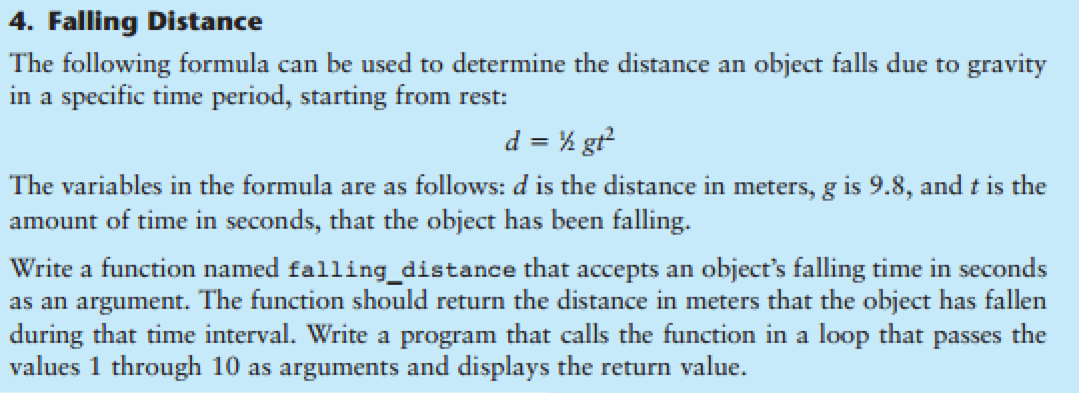


```
#3 Maximum of Two Values
def compare(intOne, intTwo):    #compares two integers and returns the one
    if intOne > intTwo:         #with the highest value
        return intOne
    else:
        return intTwo

def main():
    print("%d is the greater value." %
          compare(int(input("Enter first number: ")),
                  int(input("Enter second number: "))))

main()
```



```
>>>
Enter first number: 89
Enter second number: 99
99 is the greater value.
>>>
```

Project 4: Falling Distance (25 Pts)



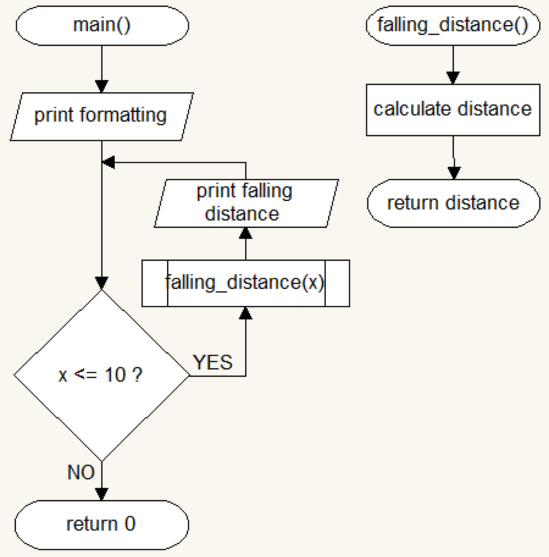
4. Falling Distance

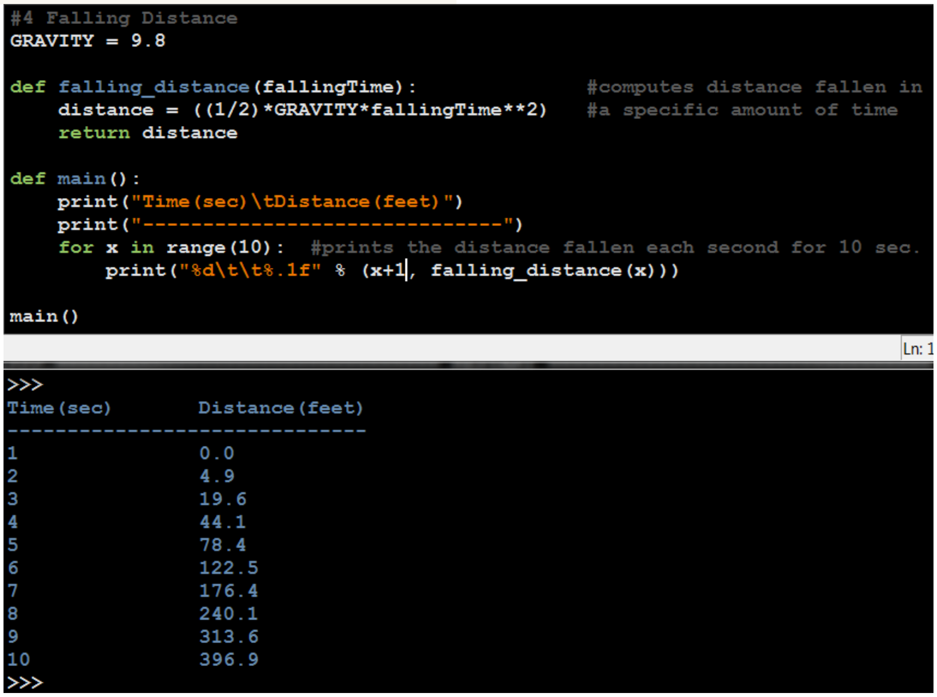
The following formula can be used to determine the distance an object falls due to gravity in a specific time period, starting from rest:

$$d = \frac{1}{2} gt^2$$

The variables in the formula are as follows: d is the distance in meters, g is 9.8, and t is the amount of time in seconds, that the object has been falling.

Write a function named `falling_distance` that accepts an object's falling time in seconds as an argument. The function should return the distance in meters that the object has fallen during that time interval. Write a program that calls the function in a loop that passes the values 1 through 10 as arguments and displays the return value.



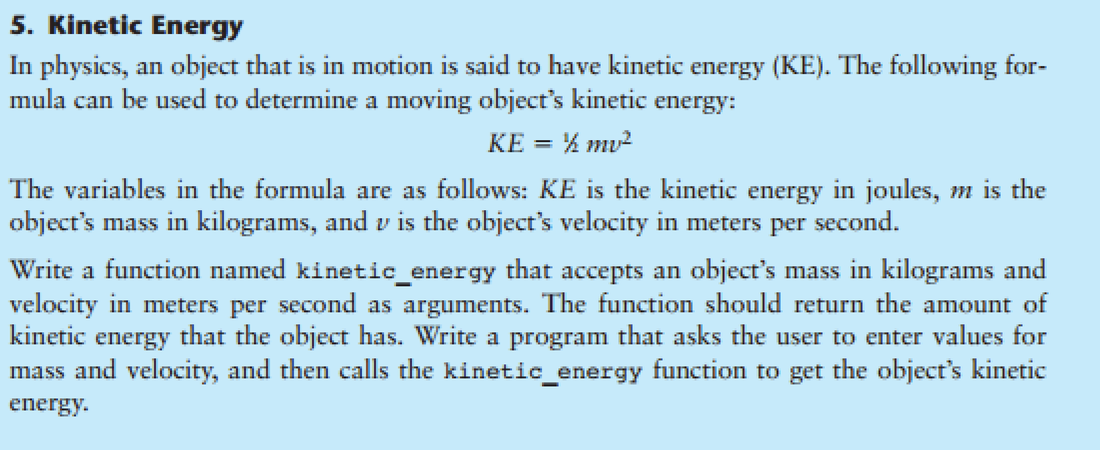


```
#4 Falling Distance
GRAVITY = 9.8

def falling_distance(fallingTime):          #computes distance fallen in
    distance = ((1/2)*GRAVITY*fallingTime**2)   #a specific amount of time
    return distance

def main():
    print("Time(sec)\tDistance(feet)")
    print("------------------------------")
    for x in range(10):  #prints the distance fallen each second for 10 sec.
        print("%d\t\t%.1f" % (x+1, falling_distance(x)))

main()
```

```
>>>
Time(sec)       Distance(feet)
------------------------------
1               0.0
2               4.9
3               19.6
4               44.1
5               78.4
6               122.5
7               176.4
8               240.1
9               313.6
10              396.9
>>>
```

Project 5: Kinetic Energy (25 Pts)

Input: Mass(kg) = 77, Velocity(m/s) = 5



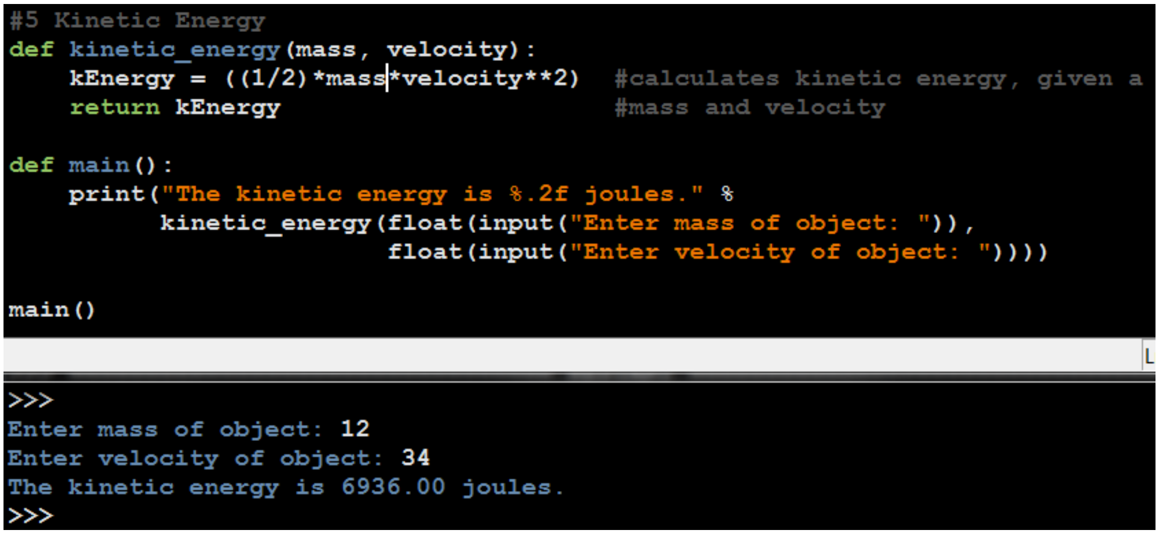
5. Kinetic Energy

In physics, an object that is in motion is said to have kinetic energy (KE). The following formula can be used to determine a moving object's kinetic energy:

$$KE = \frac{1}{2} mv^2$$

The variables in the formula are as follows: *KE* is the kinetic energy in joules, *m* is the object's mass in kilograms, and *v* is the object's velocity in meters per second.

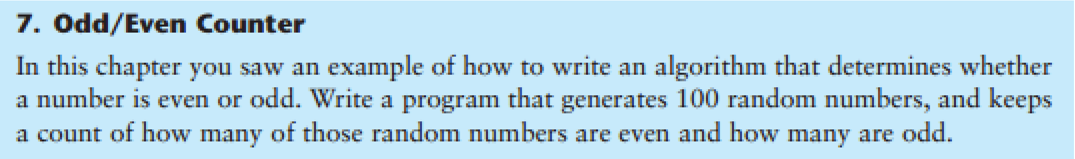
Write a function named `kinetic_energy` that accepts an object's mass in kilograms and velocity in meters per second as arguments. The function should return the amount of kinetic energy that the object has. Write a program that asks the user to enter values for mass and velocity, and then calls the `kinetic_energy` function to get the object's kinetic energy.



```
#5 Kinetic Energy
def kinetic_energy(mass, velocity):
    kEnergy = ((1/2)*mass*velocity**2)  #calculates kinetic energy, given a
    return kEnergy                      #mass and velocity

def main():
    print("The kinetic energy is %.2f joules." %
          kinetic_energy(float(input("Enter mass of object: ")),
                         float(input("Enter velocity of object: "))))

main()
```

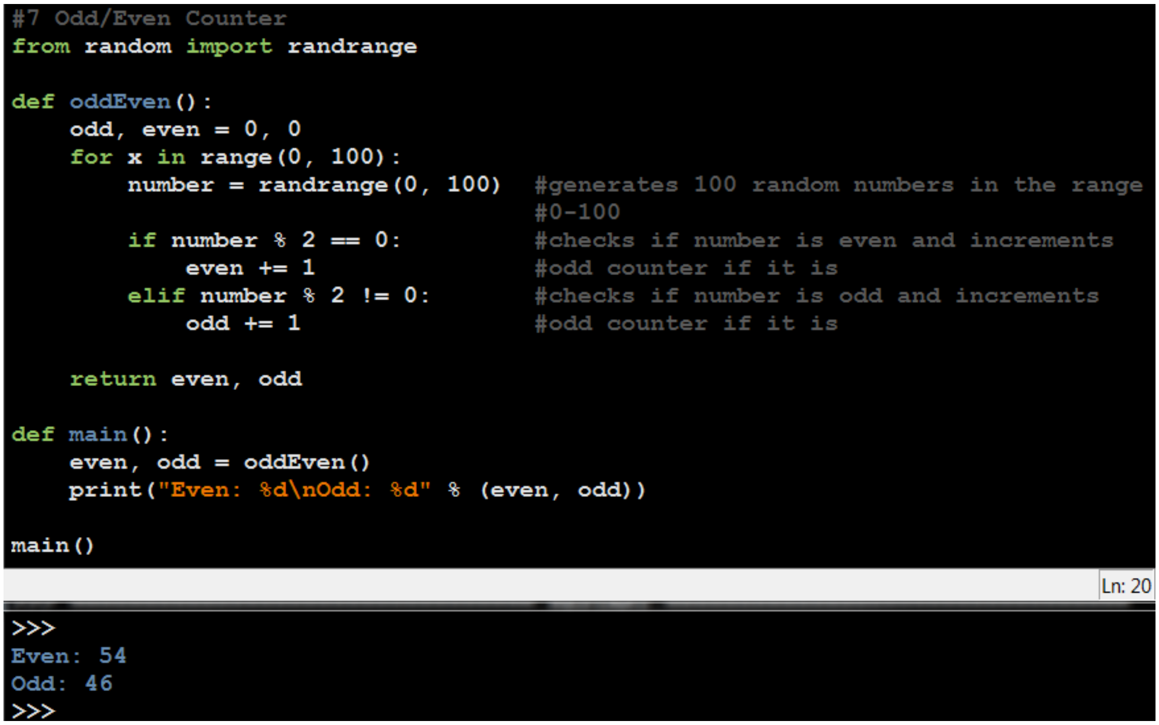
```
>>>
Enter mass of object: 12
Enter velocity of object: 34
The kinetic energy is 6936.00 joules.
>>>
```

Project 7: Odd/Even Counter (25 Pts)



7. Odd/Even Counter

In this chapter you saw an example of how to write an algorithm that determines whether a number is even or odd. Write a program that generates 100 random numbers, and keeps a count of how many of those random numbers are even and how many are odd.

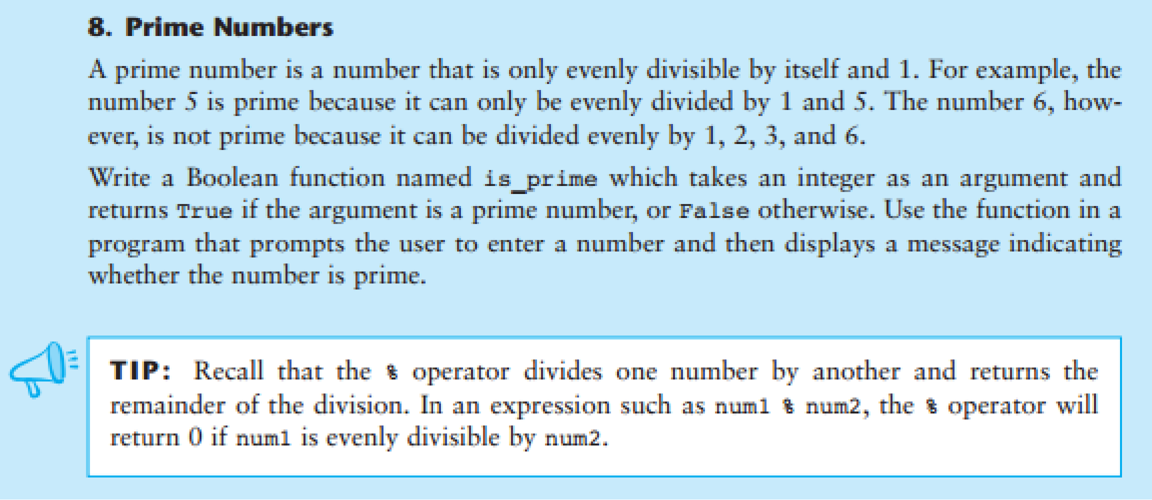


```
#7 Odd/Even Counter
from random import randrange

def oddEven():
    odd, even = 0, 0
    for x in range(0, 100):
        number = randrange(0, 100)  #generates 100 random numbers in the range
                                    #0-100
        if number % 2 == 0:         #checks if number is even and increments
            even += 1               #odd counter if it is
        elif number % 2 != 0:       #checks if number is odd and increments
            odd += 1                #odd counter if it is

    return even, odd

def main():
    even, odd = oddEven()
    print("Even: %d\nOdd: %d" % (even, odd))

main()
```

```
>>>
Even: 54
Odd: 46
>>>
```

Project 8: Prime Numbers (25 Pts)

Input: 107, 245

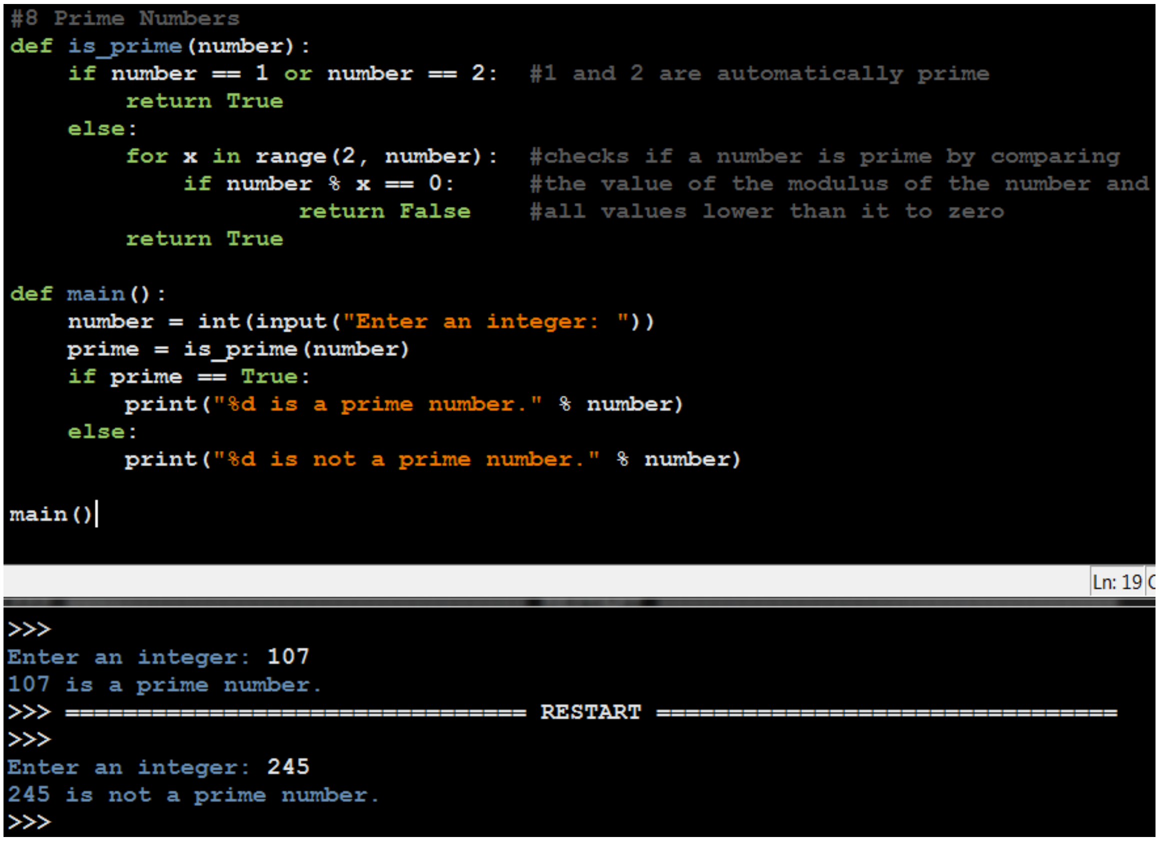


8. Prime Numbers

A prime number is a number that is only evenly divisible by itself and 1. For example, the number 5 is prime because it can only be evenly divided by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 3, and 6.

Write a Boolean function named `is_prime` which takes an integer as an argument and returns `True` if the argument is a prime number, or `False` otherwise. Use the function in a program that prompts the user to enter a number and then displays a message indicating whether the number is prime.

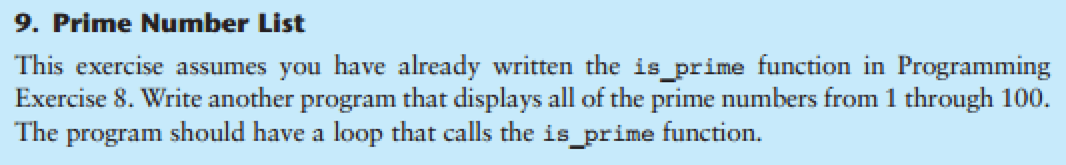
TIP: Recall that the `%` operator divides one number by another and returns the remainder of the division. In an expression such as `num1 % num2`, the `%` operator will return 0 if `num1` is evenly divisible by `num2`.



```
#8 Prime Numbers
def is_prime(number):
    if number == 1 or number == 2:  #1 and 2 are automatically prime
        return True
    else:
        for x in range(2, number):  #checks if a number is prime by comparing
            if number % x == 0:     #the value of the modulus of the number and
                    return False    #all values lower than it to zero
        return True

def main():
    number = int(input("Enter an integer: "))
    prime = is_prime(number)
    if prime == True:
        print("%d is a prime number." % number)
    else:
        print("%d is not a prime number." % number)

main()
```

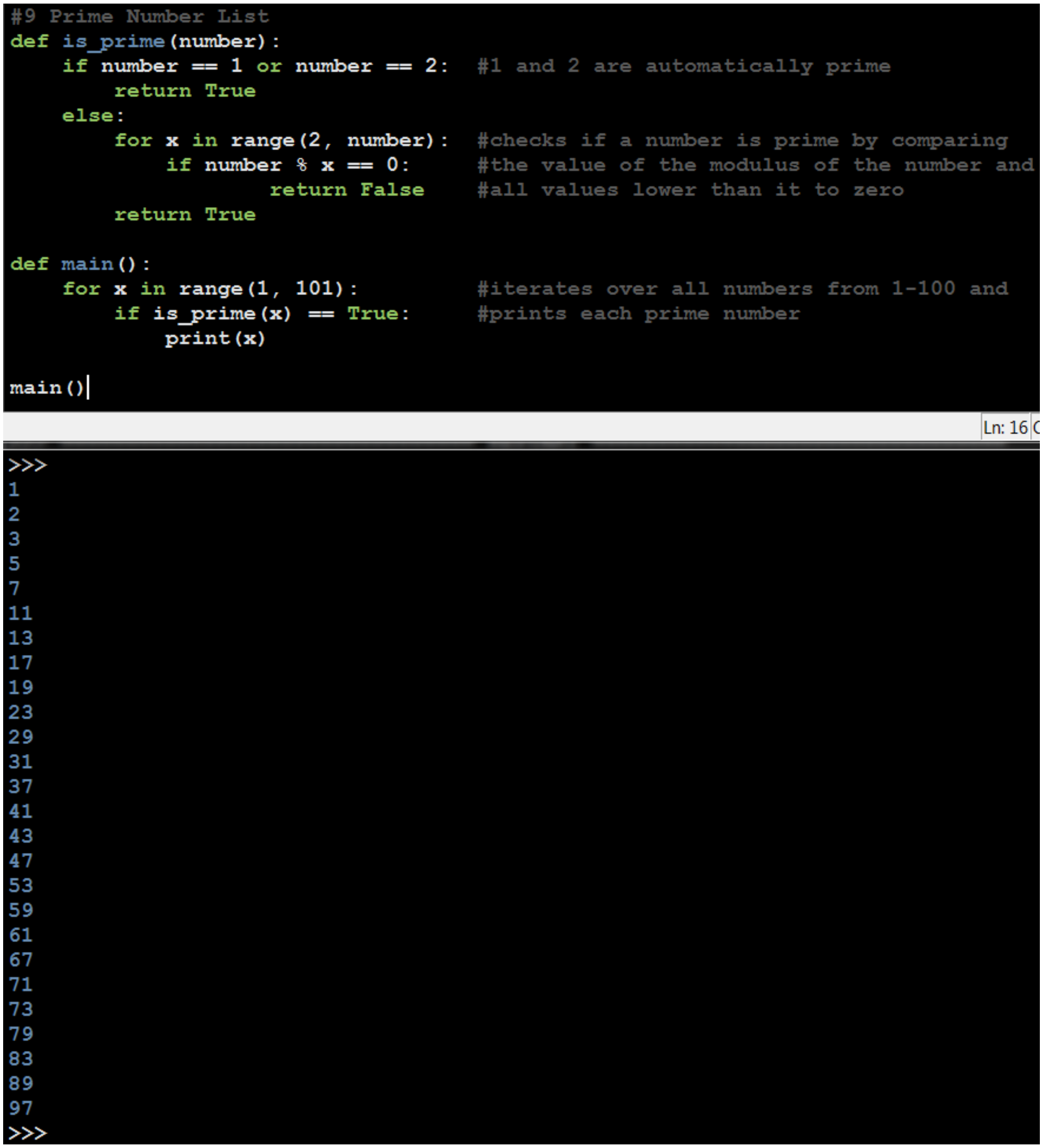
```
>>>
Enter an integer: 107
107 is a prime number.
>>> ================================ RESTART ================================
>>>
Enter an integer: 245
245 is not a prime number.
>>>
```

Project 9: Prime Number List (20 Pts)



9. Prime Number List

This exercise assumes you have already written the is_prime function in Programming Exercise 8. Write another program that displays all of the prime numbers from 1 through 100. The program should have a loop that calls the is_prime function.

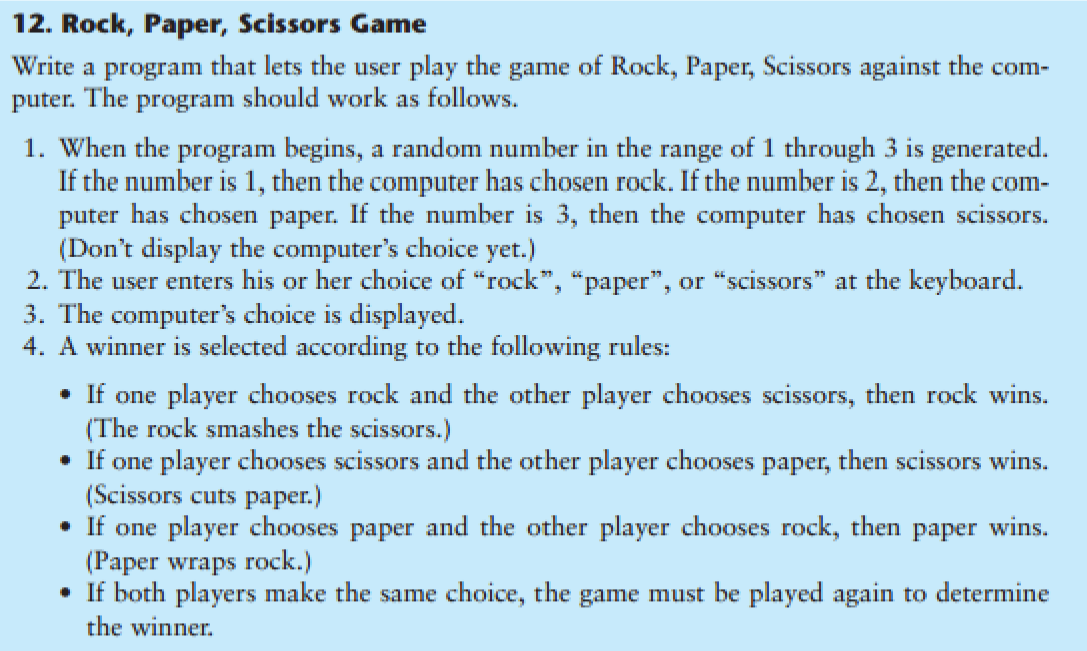


```
#9 Prime Number List
def is_prime(number):
    if number == 1 or number == 2:  #1 and 2 are automatically prime
        return True
    else:
        for x in range(2, number):  #checks if a number is prime by comparing
            if number % x == 0:     #the value of the modulus of the number and
                return False        #all values lower than it to zero
        return True

def main():
    for x in range(1, 101):         #iterates over all numbers from 1-100 and
        if is_prime(x) == True:     #prints each prime number
            print(x)

main()
```

```
>>>
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
>>>
```

Project 12: Rock, Paper, Scissors Game (60 Pts)

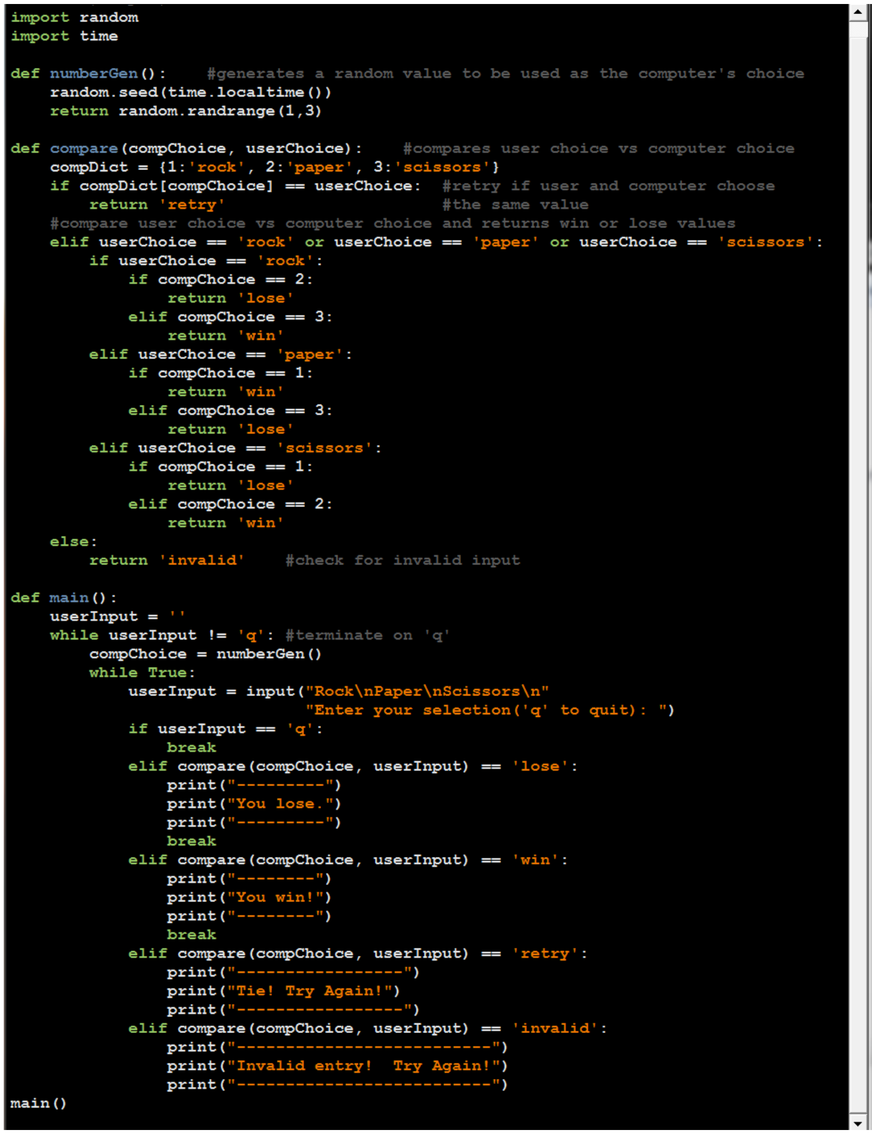
Input: = “rock”, “paper”, “scissors”



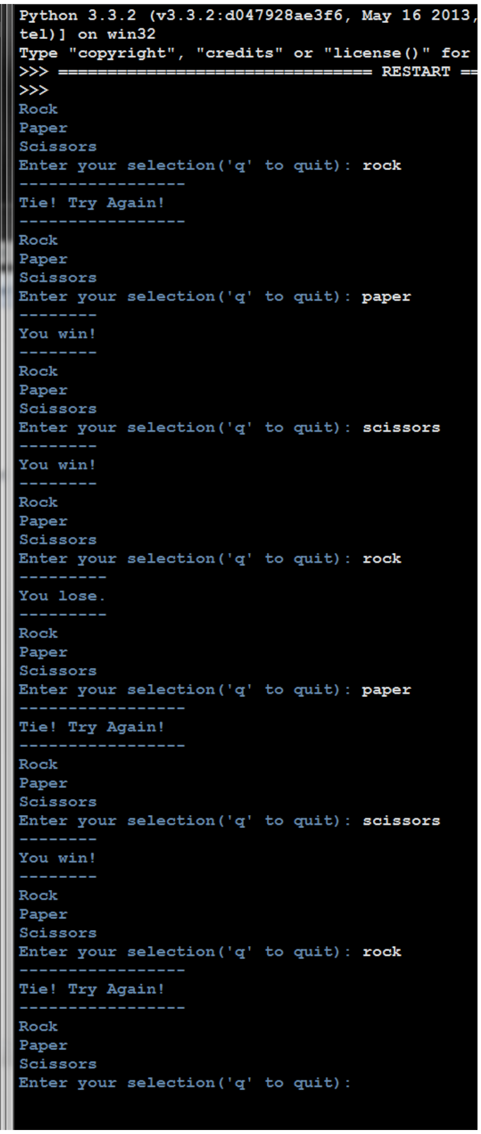
12. Rock, Paper, Scissors Game

Write a program that lets the user play the game of Rock, Paper, Scissors against the computer. The program should work as follows.

1. When the program begins, a random number in the range of 1 through 3 is generated. If the number is 1, then the computer has chosen rock. If the number is 2, then the computer has chosen paper. If the number is 3, then the computer has chosen scissors. (Don’t display the computer’s choice yet.)
2. The user enters his or her choice of “rock”, “paper”, or “scissors” at the keyboard.
3. The computer’s choice is displayed.
4. A winner is selected according to the following rules:
   - If one player chooses rock and the other player chooses scissors, then rock wins. (The rock smashes the scissors.)
   - If one player chooses scissors and the other player chooses paper, then scissors wins. (Scissors cuts paper.)
   - If one player chooses paper and the other player chooses rock, then paper wins. (Paper wraps rock.)
   - If both players make the same choice, the game must be played again to determine the winner.



```
import random
import time

def numberGen():    #generates a random value to be used as the computer's choice
    random.seed(time.localtime())
    return random.randrange(1,3)

def compare(compChoice, userChoice):    #compares user choice vs computer choice
    compDict = {1:'rock', 2:'paper', 3:'scissors'}
    if compDict[compChoice] == userChoice:  #retry if user and computer choose
        return 'retry'                      #the same value
    #compare user choice vs computer choice and returns win or lose values
    elif userChoice == 'rock' or userChoice == 'paper' or userChoice == 'scissors':
        if userChoice == 'rock':
            if compChoice == 2:
                return 'lose'
            elif compChoice == 3:
                return 'win'
        elif userChoice == 'paper':
            if compChoice == 1:
                return 'win'
            elif compChoice == 3:
                return 'lose'
        elif userChoice == 'scissors':
            if compChoice == 1:
                return 'lose'
            elif compChoice == 2:
                return 'win'
    else:
        return 'invalid'    #check for invalid input

def main():
    userInput = ''
    while userInput != 'q': #terminate on 'q'
        compChoice = numberGen()
        while True:
            userInput = input("Rock\nPaper\nScissors\n"
                              "Enter your selection('q' to quit): ")
            if userInput == 'q':
                break
            elif compare(compChoice, userInput) == 'lose':
                print("---------")
                print("You lose.")
                print("---------")
                break
            elif compare(compChoice, userInput) == 'win':
                print("--------")
                print("You win!")
                print("--------")
                break
            elif compare(compChoice, userInput) == 'retry':
                print("-----------------")
                print("Tie! Try Again!")
                print("-----------------")
            elif compare(compChoice, userInput) == 'invalid':
                print("--------------------------")
                print("Invalid entry!  Try Again!")
                print("--------------------------")
main()
```



```
Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013,
tel)] on win32
Type "copyright", "credits" or "license()" for
>>> ================================ RESTART ==
>>>
Rock
Paper
Scissors
Enter your selection('q' to quit): rock
-----------------
Tie! Try Again!
-----------------
Rock
Paper
Scissors
Enter your selection('q' to quit): paper
--------
You win!
--------
Rock
Paper
Scissors
Enter your selection('q' to quit): scissors
--------
You win!
--------
Rock
Paper
Scissors
Enter your selection('q' to quit): rock
---------
You lose.
---------
Rock
Paper
Scissors
Enter your selection('q' to quit): paper
-----------------
Tie! Try Again!
-----------------
Rock
Paper
Scissors
Enter your selection('q' to quit): scissors
--------
You win!
--------
Rock
Paper
Scissors
Enter your selection('q' to quit): rock
-----------------
Tie! Try Again!
-----------------
Rock
Paper
Scissors
Enter your selection('q' to quit):
```