1. How many times will the following for loop execute and what's the output?

   ```python
   for i in range(-1, 7, -2):
       for j in range(3):
           print(1, j)
   ```

   Prints nothing since loop is not executed at all (Range is not valid)

2) How many times will the following for loop execute and what's the output?

   ```python
   for i in range(1, 3, 1):
       for j in range(i+1):
           print('*')
   ```

   Output:
   ```
   *
   *
   *
   *
   ```
3. What does the following code produce?

```python
for i in range(-1, 2, 1):
    for j in range(0, 3, 1):
        print(i + j)

print()
```

**output:**

```
-1
0
1
0
1
2
1
2
3
```
3.2) What does the following code produce?

```python
for i in range(0, 5, 1):
    for j in range(0, i, 1):
        print(i)
        print()
```

output:

```
1
2
2
2
2
3
3
3
3
3
4
4
4
4
4
```
4. Functions short questions (see readings and book notes for answers for these questions):

a) What is the difference between a user-defined and a built-in function?

**User defined function** is a function defined the program that you are using. It should include the function that’s being defined.

```python
def func():
    print("hello")
def main():
    func()
main()
```

**Built-in function:** a function that has been defined and you just use it in your program:

```python
print('hi')
```

b) What is the difference between a local and a global variable? Give an example?

**Local variable**
A variable defined inside a function. A local variable can only be used inside its function. Parameters of a function are also a special kind of local variable.

```python
def fun(num):
    print(num)
```

**Global variable:** is a variable declared outside of all functions and would be visible and can be used in all functions.

```python
Num= 20

def fun():
    print(num)
```

c) Explain call-by-value?
Call by value is when you pass a value of a variable to a function

```python
func(num)
```

d) What is an argument and a parameter and give an example for each?

**Argument**
A value provided to a function when the function is called. This value is assigned to the corresponding parameter in the function.

The argument can be the result of an expression which may involve operators, operands and calls to other fruitful functions.

**parameter**

A name used inside a function to refer to the value which was passed to it as an argument.

Example of an argument:
```
func(arg)
```

**Example of parameter:**
```
def func(parm):
```

e) How to do you return a value or values? give an example?

You use the keyword return in function and then you need to assign a variable in the function call when you invoke a function:

```
def func():
    greeting = “hello”
    return greeting # use return to return value
def main():
    message = func() # function call
main()
```

4) String questions:
a) provide a list of 6 string methods and what they do and example for each?

```
name = ‘cat’
name.title()
name.find(‘e’)
name.split()
name.revers()
name.replace(‘o’, ‘a’)
name.count(‘i’)
```

b) Write one sentence to do the following:
1) Count the number of ’t’ in a sentence
```
sentence.count(‘t’)
```
2) find if "the" is in the sentence or not?

if (sentence.find('the') != -1)

3) Replace all the "o" in the sentence with space.
   sentence.replace('o', ' ')

5) reverse the sentence.
   sentence.reverse()
5)
Write a program that reads an integer \( N \) from the keyboard and then calls a user defined function to compute and displays the sum of the numbers from \( N \) to \( (2 \times N) \) if \( N \) is nonnegative. If \( N \) is a negative number, then it's the sum of the numbers from \( (2 \times N) \) to \( N \). The starting and ending points are included in the sum.

Sample run:

Enter number : 5
Sum : 45

Sample run:

Enter number : -4
Sum : -30

Notes and hints:
• You should have only 1 loop to compute the sum; use the sign of \( N \) to determine the start point and the end point.

def total(l, h):
    tot = 0
    for i in range(l, h+1, 1 ):
        tot += i
    print("Sum: ", tot);

def main():
    n = int(input("Enter number:"))
    if (n > 0):
        # if n is greater than zero
        low = n
        high = 2 * n
        total(low, high)
    else:
        # if n is a negative number
        low = 2 * n
        high = n
        total(low, high)

    main()
6.

Write a program that reads a date as an integer in the format MMDDYYYY. The program will call a function that prints print out the date in the format <Month Name> <day>, <year>.

Sample run:

Enter date: 12252003

In the input the month and the day will always be given as two digits, including leading zero. So the 20th of May, 2000 will be entered as 05202000 and the 3rd of December, 2000 will be entered as 12032000.

Don’t worry about writing down all of the months, write code that can handle just the first couple and put a comment like:

# Remaining months the same way

```python
def printFunc(year, month, day):
    if (month == 1):
        print("January, end=")

    elif (month == 2):
        print("February , end=")

    elif (month == 3):
        print("March ", end="")

    elif (month == 4):
        print("April" , end="")

    elif (month == 5):
        print("May", end="")

    elif (month == 6):
        print("June" , end="")

    elif (month == 7):
```

print("July ", end=")

elif (month == 8):
    print("August", end=")

elif (month == 9):
    print("September", end=")

elif (month == 10):
    print("October", end=")

elif (month == 11):
    print("November", end=")

elif (month == 12):
    print("December", end=")

else:
    print("Invalid month")

print("", day, ", ", year, ".")

def main():
    date = int(input("Enter an integer in the format MMDDYYYY:"))
    month = date // 1000000
    day = date % 1000000 // 10000
    year = date % 10000
    printFunc(year, month, day)
main()
7. Write a program that reads a positive integer \( N \) from the keyboard and call a user defined function outputs all of the prime numbers from 1 up to \( N \). Write and use a function:

```python
def isPrime (num):
    # assumes num is positive
    # limit to check for sqrt when checking for primes
    limit = int(math.sqrt(num))

    if ((num == 1) or (num == 2)):
        # known primes
        return True
    else:
        for i in range(2, limit +1, 1):
            if ((num % i) == 0):
                # Not prime since its divisable by i
                return False

    # is prime
    return True
```

The main function will prompt for the number \( N \) and then it will test all of the numbers from 1 to \( N \) inclusive calling \texttt{isPrime} for each number. If the number is prime it will print it.

**Note:** A prime number is a number divisible only by itself and 1. Otherwise, it's a composite!

Sample run:

```
Enter number : 9
Prime numbers: 1 2 3 5 7
```
def main():

    n = int(input("Enter number: "))

    for i in range(n+1):
        if isPrime(i):
            print (i , " Is Prime ")
        else:
            print (i , " NOT Prime ")

main()

8. Write a program that will pass temperatures to a function that will add 5 degrees to each of the temperatures (making them hotter) and return the new temperature.

# Call a function to make NYC weather warmer by 10 degrees

def makehot(tem):
    tem = tem + 5
    print("n Printing in make hot function after adding 10 degrees:")
    print(tem)
    return

def main():
    temp = float(input('Enter temp for NYC: '))

    #call a user-defined function to make temperatures hot by 5 degrees
    makehot(temp)

main()
9. Write a program that will generate 5 passwords (made up of one capital letter, followed by 6 small letters, followed by one number). The range of the number also should be selected by the user. Use one function to generate all of the letters and numbers randomly. Print the generated passwords.

import random

def makepassword():
    password = ""
    for letter in range(1,9,1): # Create each password of length 8 characters
        if (letter == 1):
            #Generate a capital letter (first letter should be capital)
            password = password + random.choice('ABCDEFGHIJKLMNOPQRSTUVXYZ')
        elif (letter == 8):
            #Generate a number (last character should be a number)
            password = password + random.choice('123456789')
        else:
            # 2nd to 7th letter should be small case
            password = password + random.choice('abcdefghijklmnopqrstuvwxyz')

    return password

def main():
    # generate 5 different passwords randomly
    for i in range(5):
        newpass = makepassword()
        print ("Password # ", i, newpass)

main()
10) Write a module to print a square, triangle, and rectangle. Write another program with a main function that imports that shapes module based on the user choice of height, width and character to use to print the shape.

shapes.py module:

# shapes.py module to create shapes... no main function needed

"""A collection of functions
for printing basic shapes. This will be used as part of documentation for this module.
try to type on the Interpreter  print(shapes.__doc__)""

CHAR = '*'

def rectangle(height, width):
    """ Prints a rectangle."
    for row in range(height):
        for col in range(width):
            print(CHAR, end = '')
        print()

    print()

def square(side):
    """ Prints a square."
    rectangle(side, side)

def triangle(height):
    """ Prints a right triangle."
    for row in range(height):
        for col in range(1, row + 2):
            print(CHAR, end = '')
        print()

    print()
# main.py: program with main function to import and use shapes module

import shapes

def main():
    # draw a square
    width = int(input("Enter width of square: "))
    height = int(input("Enter height of square: "))

    # draw a square by invoking shapes module function square()
    shapes.square(height)

    # draw a triangle by invoking shapes module function triangle()
    shapes.triangle(height)

    # draw a rectangle by invoking shapes module function rectangle()
    shapes.rectangle(height, width)

main()