1.Write a Python program to implement Magic Square

A magic square of order n is an arrangement of n^2 numbers, usually distinct integers, in a square, such that the n numbers in all rows, all columns, and both diagonals sum to the same constant. A magic square contains the integers from 1 to n^2.

**The constant sum in every row, column and diagonal is called the magic constant or magic sum, M. The magic constant of a normal magic square depends only on n and has the following value:  
M = n(n^2+1)/2  
example  
Magic Square of size 5**

9 3 22 16 15  
2 21 20 14 8  
25 19 13 7 1  
18 12 6 5 24  
11 10 4 23 17  
Sum in each row & each column = 5\*(5^2+1)/2 = 65

def generateSquare(n):

magicSquare = [[0 for x in range(n)]

for y in range(n)]

i = n / 2

j = n - 1

num = 1

while num <= (n \* n):

if i == -1 and j == n:

j = n – 2

i = 0

else:

if j == n:

j = 0

if i < 0:

i = n - 1

if magicSquare[int(i)][int(j)]:

j = j - 2

i = i + 1

continue

else:

magicSquare[int(i)][int(j)] = num

num = num + 1

j = j + 1

i = i - 1

print ("Magic Square for n =", n)

print ("Sum of each row or column",n \* (n \* n + 1) / 2, "\n")

for i in range(0, n):

for j in range(0, n):

print('%2d ' % (magicSquare[i][j]),end = '')

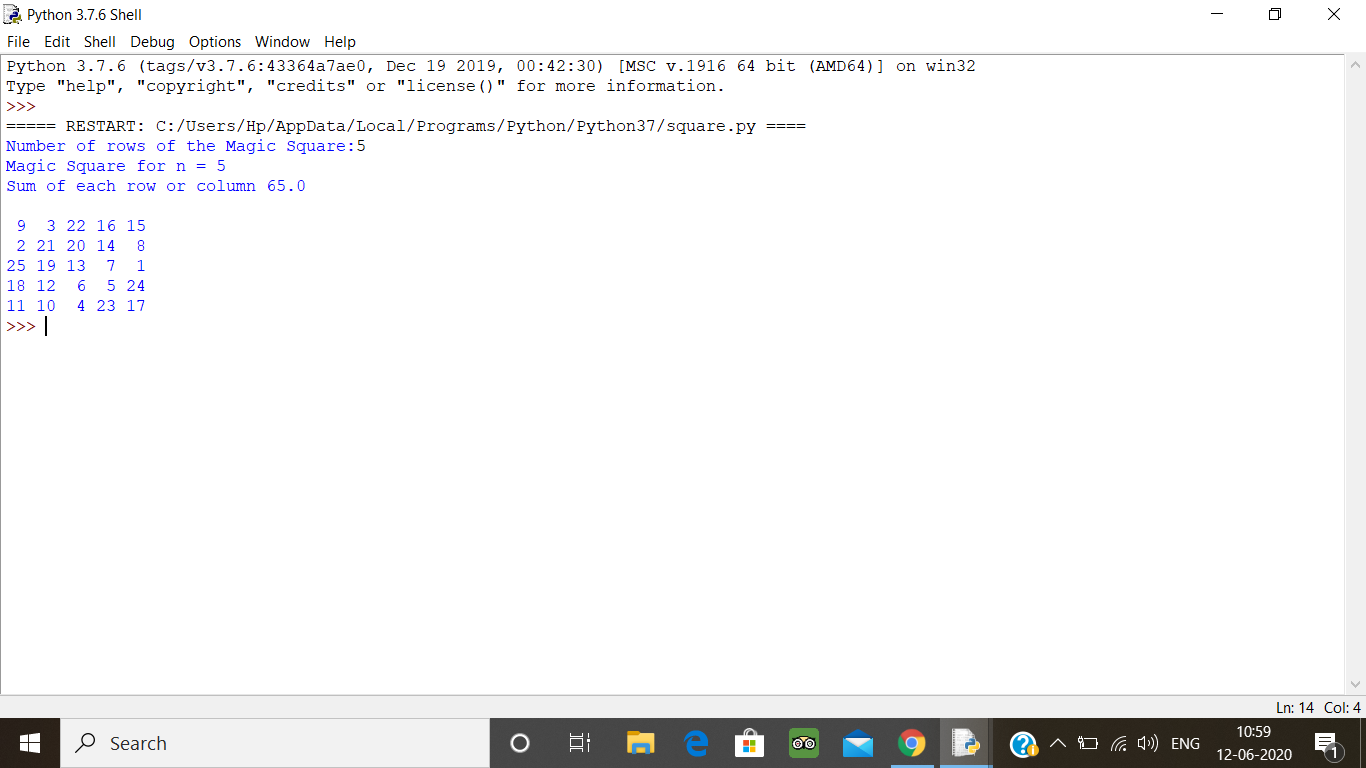
if j == n - 1:

print()

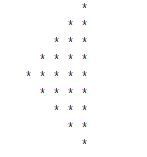
n=int(input("Number of rows of the Magic Square:"))

generateSquare(n)

**output:**



2. Python program to print the pattern



def pattern(n):

    k = 2 \* n - 2

    for i in range(0, n-1):

        for j in range(0, k):

            print(end=" ")

        k = k - 2

        for j in range(0, i + 1):

            print("\* ", end="")

        print("")

    k = -1

    for i in range(n-1,-1,-1):

        for j in range(k,-1,-1):

            print(end=" ")

        k = k + 2

        for j in range(0, i + 1):

            print("\* ", end="")

        print("")

pattern(5)

**output:**

