

**REPORT FOR**  
**Python Mini Project**  
**AS A PROJECT WORK FOR THE COURSE**  
**PYTHON PROGRAMMING (INT 108)**

**NAME:** Ayush Raj

**REGISTRATION NUMBER:** 12216328

**ROLL NUMBER:** A10

**SEMESTER:** FIRST SEMESTER

**DATE OF SUBMISSION:** 31/12/22

**SUBMITTED TO:** Ms. Manbir Kaur



**L** OVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

---

*Transforming Education Transforming India*

**SCHOOL OF COMPUTER SCIENCE AND  
ENGINEERING**

**LOVELY PROFESSIONAL  
UNIVERSITY, JALANDHAR, PUNJAB,  
INDIA.**

# **DECLARATION**

This is to declare that this report has been written by me. No part of this report is copied from other sources. All information included from other source has been duly acknowledged. I aver that if any part of the report is found to be copied. I shall take full responsibility for it.

**Ayush Raj**

**12216728**

# INDEX

1.INTRODUCTION

2.LOOPS/FUNCTION

3.PROJECT

4.CODE

5.RESULT

# 1.INTRODUCTION:

This is a program that generates a magic square of size ' $n$ '. A magic square is a square grid of numbers such that the sum of the numbers in each row, column, and diagonal is the same. The program prompts the user to input the size of the magic square, and then generates and prints out the magic square using a specific algorithm.

The algorithm works by starting at the middle of the top row and placing the first number. It then moves to the next position according to a specific set of rules, and continues this process until all of the cells in the grid have been filled with numbers. If the next position is already occupied, the algorithm skips that position and moves to the next one according to the rules.

After the magic square has been generated, the program prints out the sum of the numbers in each row or column, which should be the same for all rows, columns, and diagonals.

## 2.LOOPS/FUNCTION:

The program uses a **'while'** loop to generate the magic square. The loop continues until all of the cells in the grid have been filled with numbers.

The program also uses a nested **'for'** loop to print out the magic square. The outer **'for'** loop iterates through the rows of the square, and the inner **'for'** loop iterates through the columns of each row.

The program also defines a function called **'matrix'** which takes in a single argument **'n'** which is the size of the magic square. The function generates and prints out the magic square using the algorithm described above.

# 3.PROJECT

A basket is given to you in the shape of a matrix. If the size of the matrix is  $N \times N$  then the range of number of eggs you can put in each slot of the basket is 1 to  $N^2$ . Your task is to arrange the eggs in the basket such that the sum of each row, column and the diagonal of the matrix remain same.

Test case for your reference:

Input by the user dimension of the basket i.e.,  $N = 3$

So, number of eggs you can put at each slot are in the range of 1 to  $3^2$  (1 to 9)

**Input:**

6 3 6

5 5 5

4 7 4

**Explanation:**

Now the value of the sum of  
any row or column as well as diagonal is 15

Note:  $2 \leq N \leq 100$

## 4. CODE:

```
def matrix(n):
    m = [[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 m[int(i)][int(j)]:
            j = j - 2
            i = i + 1
            continue
        else:
            m[int(i)][int(j)] = num
            num = num + 1
        j = j + 1
        i = i - 1
    print ("Sum of eggs in each row or column and diagonal ",n * (n * n + 1) / 2,
"\n")
    for i in range(0, n):
        for j in range(0, n):
            print('%2d ' % (m[i][j]),end = ")
            if j == n - 1:
                print()
    n=int(input("Number of rows of matrix:"))
    matrix(n)
```

# 5. RESULTS:

```
>>> Python 3.10.7 (v3.10.7:6cc6b13308, Sep  5 2022, 14:02:52) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: /Users/ayushraj/Documents/ayushminiproject.py =====
Number of rows of matrix:1
Sum of eggs in each row or column and diagonal  1.0

1
Number of rows of matrix:3
Sum of eggs in each row or column and diagonal  15.0

2  7  6
9  5  1
4  3  8

>>> ===== RESTART: /Users/ayushraj/Documents/ayushminiproject.py =====
Number of rows of matrix:5
Sum of eggs in each row or column and diagonal  65.0

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
Number of rows of matrix:7
Sum of eggs in each row or column and diagonal  175.0

20  12  4  45  37  29  28
11  3  44  36  35  27  19
2  43  42  34  26  18  10
49  41  33  25  17  9  1
40  32  24  16  8  7  48
31  23  15  14  6  47  39
22  21  13  5  46  38  30

>>> ===== RESTART: /Users/ayushraj/Documents/ayushminiproject.py =====
```