## GROUP A-07:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

## PROGRAM:---

```
# GROUP A - 7 Write a python Program for magic square. A magic square is an
arrangemen0t of the numbers
# from 1 to N^2 (N-squared) in an NxN matrix, with each number occurring
exactly once, and such that the
# sum of the entries of any row, any column, or any main diagonal is the
same. Perform follwing operations
# a) Generate Magic Square
# b) Check whether matrix is magic square or not

# Generate odd sized magic squares


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 Squre for n =", n)
    print("Sum of each row or column or diagonal i.e Magic Number is :
{0:.0f}".format(n * (n * n + 1) / 2), "\n")
    print_mat(magicsquare, n)


# Printing magic square

def print_mat(t, n):
    for i in range(0, n):
        for j in range(0, n):
```

```
            print(t[i][j], end=" ")
        print()


# Determine whether a given matrix is magic matrix or not

def isMagicSquare(mat, n):
    s = 0                           # calculate the sum of the prime diagonal
    for i in range(0, n):
        s = s + mat[i][i]

    s2 = 0                          # Calculate the sum of the secondary
diagonal
    for i in range(0, n):
        s2 = s2 + mat[i][n - i - 1]
    if (s != s2):
        return False
    for i in range(0, n):           # For sums of Rows
        rowsum = 0;
        for j in range(0, n):
            rowsum += mat[i][j]
        if (rowsum != s):           # check if every row sum is equal to prime
diagonal sum
            return False
    for i in range(0, n):           # For sums of Columns
        colsum = 0
        for j in range(0, n):
            colsum += mat[j][i]
        if (s != colsum):           # check if every column sum is equal to
prime diagonal sum
            return False
    return True


flag = 1

while flag == 1:
    menu = " /~~~~~~~~~~MENU~~~~~~~~~~/ \n" \
           "1. Generate Magic Square \n" \
           "2. Determine whether matrix is magic square or not \n" \
           "3. Exit"

    print(menu)
    choice = int(input("Enter your choice : "))
    if choice == 1:
        n = int(input(" Enter the size of Magic square : "))
        generatesquare(n)
    elif choice == 2:
        n = int(input(" Enter the size of Magic square : "))


# Accept the matrix of size (n X n) and Initialize matrix
        mat = []
        print("Enter the elements rowwise:")


# For user input
```

```
        for i in range(0, n):  # A for loop for row entries
            a = []
            for j in range(0, n):  # A for loop for column entries
                a.append(int(input("Enter element : ")))
            mat.append(a)
        if (isMagicSquare(mat, n)):
            print("Magic Square")
        else:
            print("Not a magic Square")
    else:
        print("Wrong Choice Please Choose Another Option ")
        flag = 0
```

# OUTPUT:---

/~~~~~~~~~~MENU~~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 1

Enter the size of Magic square : 5

Magic Squre for n = 5

Sum of each row or column or diagonal i.e Magic Number is : 65

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

/~~~~~~~~~~MENU~~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 2

Enter the size of Magic square : 5

Enter the elements rowwise:

Enter element : 1

Enter element : 2

Enter element : 3

Enter element : 4

Enter element : 5

Enter element : 6

Enter element : 7

Enter element : 8

Enter element : 9

Enter element : 10

Enter element : 11

Enter element : 12

Enter element : 13

Enter element : 14

Enter element : 15

Enter element : 16

Enter element : 17

Enter element : 18

Enter element : 19

Enter element : 20

Enter element : 21

Enter element : 22

Enter element : 23

Enter element : 24

Enter element : 25

Not a magic Square

/~~~~~~~~~~MENU~~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 2

Enter the size of Magic square : 5

Enter the elements rowwise:

Enter element : 9

Enter element : 3

Enter element : 22

Enter element : 16

Enter element : 15

Enter element : 2

Enter element : 21

Enter element : 20

Enter element : 14

Enter element : 8

Enter element : 25

Enter element : 19

Enter element : 13

Enter element : 7

Enter element : 1

Enter element : 18

Enter element : 12

Enter element : 6

Enter element : 5

Enter element : 24

Enter element : 11

Enter element : 10

Enter element : 4

Enter element : 23

Enter element : 17

Magic Square

/~~~~~~~~~MENU~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 3

Wrong Choice Please Choose Another Option

Process finished with exit code 0