## Magic Squares

Square nxn number 1..n2 all sums (row, colums, diagonals) are equals = Charcteristic Number

(dac so)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MS3 | | |  | MS5 | | | | |  | MS4 | | | |  | | MS6 | | | | | | |  | |  | |
| 8 | 1 | 6 |  |  |  | 1 |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  | |  | | C = (1+..+n2)/n  = (n2+1)\*n2 / (2n)  = (n2+1)\*n / 2  C2 = 15, C4 = 35  C5 = 65, C6 = 111 | |
| 3 | 5 | 7 |  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  | |  | |
| 4 | 9 | 2 |  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  | |  | |
|  | | |  |  | | | | |  |  | | | |  | |  | | | | | | |  | |  | |

|  |  |
| --- | --- |
| Algorithm OMS (Odd Magic Squere) | A blue star with black text  Description automatically generated |
| 1. set all 0 to a  2. write 1 at (0, n/2)  3. for k = 2..n2:  3.1 Find cell a(i,j)  3.2 set a(i,j) = k  Go to direction North-East (NE) |

### Odd Magic Square (OMS)

Program

# Magic Squeres

MN = 100

BL = chr(32)

def Go(msg = ' ? '):

if input(msg) == '.': exit(0)

def Print(a, msg = ''):

v = len(str(nn)) + 1

print(msg)

for i in range(n):

print(' ')

for j in range(n):

e = str(a[i][j])

print(BL\*(v-len(e)) + e, end='')

def MS(size):

global n, nn

n = size

nn = n\*n

print('Magic Square of ', n)

if n < 1 or n == 2:

print('No solutions')

return

if n == 1:

print([1])

return

if n % 2 == 1: OddMS(n)

else: EvenMS(n)

Test()

def OddMS(n):

global a

print('Odd Magic Squere')

# Init a

a = []

for i in range(n): a.append([0]\*n)

# Print(a, '\n Init a:')

i, j = 0, n//2

a[i][j] = 1

for k in range(2, nn+1):

# tim vi tri (i,j) dat k

# theo huong DongBac

i -= 1; j += 1

if i<0 and j==n: i += 2; j -= 1 # den duoi

else:

if i < 0: i = n-1

if j == n: j = 0

if a[i][j] > 0: i += 2; j -= 1

a[i][j] = k

Print(a, '\n Result a:')

def EvenMS(n):

print('Even Magic Squere')

def Test():

s = (nn+1)\*n // 2 # dac so

#print('\n Dac so', s)

d1 = d2 = er = 0

r = range(n)

for i in r:

col = row = 0

d1 += a[i][i]; d2 += a[i][n-1-i]

for j in r:

col += a[i][j]

row += a[j][i]

if col != s: print(' ERROR in col ', i); er += 1

if row != s: print(' ERROR in row ', i); er += 1

if d1 != s: print(' ERROR in diagonal 1'); er += 1

if d2 != s: print(' ERROR in diagonal 2'); er += 1

print('\n Total Errors: ', er)

# APPLICATION

for n in range(1, 20, 2):

MS(n)

Go()

print('\n T h e E n d .')

### Even Magic Square (EMS)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |  | tạo xâu mẫu s  len = n2 = n/2  k = n2/2 kí tự T  nếu n2 lẻ thêm DN  Têm # cho đủ len = n2 | n = 6, n2 = 6/2 = 3 = len  k = n2/2 = 3/2 = 1  s = TDN |
| 7 | 8 | 9 | 10 | 11 | 12 |  |
| 13 | 14 | 15 | 16 | 17 | 18 |  |
| 19 | 20 | 21 | 22 | 23 | 24 |  |
| 25 | 26 | 27 | 28 | 29 | 30 |  |
| 31 | 32 | 33 | 34 | 35 | 36 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 36 | 5 | 33 | 4 | 2 | 31 | TDN |  |
| 25 | 8 | 9 | 10 | 11 | 12 | NTD |
| 13 | 14 | 15 | 16 | 17 | 18 | DNT |
| 19 | 20 | 21 | 22 | 23 | 24 |  |
| 7 | 26 | 27 | 28 | 29 | 30 |  |
| 6 | 32 | 3 | 34 | 35 | 1 |  |

**Program**

# Magic Squeres

MN = 100

BL = chr(32)

def Go(msg = ' ? '):

if input(msg) == '.': exit(0)

def Print(a, msg = ''):

v = len(str(nn)) + 1

print(msg)

for i in range(n):

print(' ')

for j in range(n):

e = str(a[i][j])

print(BL\*(v-len(e)) + e, end='')

def MS(size):

global n, nn, n1

n = size

nn = n\*n

n1 = n-1

print('Magic Square of ', n)

if n < 1 or n == 2:

print('No solutions')

return

if n == 1:

print([1])

return

if n >= MN:

print('Large size', n)

return

if n % 2 == 1: OddMS()

else: EvenMS()

Test()

def OddMS():

print('Odd Magic Squere')

r = range(n)

for i in r:

for j in r: a[i][j] = 0

i, j = 0, n//2

a[i][j] = 1

for k in range(2, nn+1):

# tim vi tri (i,j) dat k

# theo huong DongBac

i -= 1; j += 1

if i<0 and j==n: i += 2; j -= 1 # den duoi

else:

if i < 0: i = n-1

if j == n: j = 0

if a[i][j] > 0: i += 2; j -= 1

a[i][j] = k

Print(a, '\n Result a:')

def EvenMS():

print('Even Magic Squere')

# Init

r = range(n)

k = 1

for i in r:

for j in r:

a[i][j] = k

k += 1

Print(a, 'Init a:')

n2 = n // 2; n4 = n2 // 2

form = 'C'\*n4 # n4 C

#print(form)

if n2 % 2 == 1: form += 'VH' # neu n2 le: them HV

# dien B cho du len(form) = n2

form += 'B'\*(n2-len(form))

r1 = range(n2)

for i in r1:

print('\n form: ', form)

for j in r1:

if form[j] == 'C': Center(i,j); continue

if form[j] == 'V': Vertical(i,j); continue

if form[j] == 'H': Horizontal(i,j); continue

# rotate form by clockwise

form = form[-1] + form[:-1]

Print(a, '\n Result a:')

def Center(i,j): # doi xung cheo

a[i][j], a[n1-i][n1-j] = a[n1-i][n1-j], a[i][j]

a[n1-i][j], a[i][n1-j] = a[i][n1-j], a[n1-i][j]

def Vertical(i,j): # doi xung doc

a[i][j], a[i][n-1-j] = a[i][n-1-j], a[i][j]

def Horizontal(i,j): # doi xung nagng

a[i][j], a[n-1-i][j] = a[n-1-i][j], a[i][j]

def Test():

c = (nn+1)\*n // 2 # dac so

print('\n Dac so', c)

d1 = d2 = er = 0

r = range(n)

for i in r:

col = row = 0

d1 += a[i][i]; d2 += a[i][n-1-i]

for j in r:

col += a[i][j]

row += a[j][i]

if col != c: print(' ERROR in col ', i); er += 1

if row != c: print(' ERROR in row ', i); er += 1

if d1 != c: print(' ERROR in diagonal 1'); er += 1

if d2 != c: print(' ERROR in diagonal 2'); er += 1

print('\n Total Errors: ', er)

def Init(): # Khoi tao a[100][100

global a

a = []

for i in range(MN): a.append([0]\*MN)

# APPLICATION

Init()

for n in range(20):

MS(n)

Go()

print('\n T h e E n d .')