## 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 |  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  | |  | | 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

/\*

Name: MS.CPP

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Magic Square

'''

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 100;

int a[MN][MN];

int maxlen;

const string DIGIT = "0123456789";

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

string Str(int n) {

if(n == 0) return "0";

string s = "";

while(n) {

s = DIGIT[n % 10] + s;

n /= 10;

}

return s;

}

void Print(int a[], int n, const char \* msg = "") {

cout << msg;

for (int i = 0; i < n; ++i) {

string s = Str(a[i]);

for(int j = s.length(); j <= maxlen; ++j) {

s = " " + s;

}

cout << " " << s;

}

}

void Print(int a[][MN], int n, const char \* msg = "") {

cout << msg;

for (int i = 0; i < n; ++i) {

Print(a[i], n, "\n ");

}

}

void Test(int n) {

int c = (n\*n+1)\*n/2;

int col, row;

int d1 = 0, d2 = 0;

int er = 0;

for(int i = 0; i < n; ++i) {

col = row = 0;

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

for(int j = 0; j < n; ++j) {

col += a[i][j];

row += a[j][i];

}

if(col != c) {

cout << "\n ERROR in row " << i;

++er;

}

if(row != c) {

cout << "\n ERROR in col " << i;

++er;

}

}

if(d1 != c) {

cout << "\n ERROR in diagonal 1";

++er;

}

if(d2 != c) {

cout << "\n ERROR in diagonal 2";

++er;

}

if(er == 0) cout << "\n CORRECT.";

}

void OMS(int n) {

for(int i = 0; i < n; ++i)

for(int j = 0; j < n; ++j)

a[i][j] = 0;

int i = 0, j = n/2;

a[i][j] = 1;

int nn = n\*n;

for(int k = 2; k <= nn; ++k) {

// (i,j) ?

--i; ++j;

if(i < 0 && j == n) {

// below (i,j)

i += 2; --j;

}

else if(i < 0) i = n-1;

else if(j == n) j = 0;

if(a[i][j] > 0) {

i += 2; --j;

}

a[i][j] = k;

}// for

Print(a,n,"\n Result: \n");

Test(n);

}

void EMS(int n) {

cout << "\n Will be released.";

}

void MS(int n) {

cout << "\n Magic Square of " << n;

if(n < 1 || n == 2) {

cout << "\n n < 0 or n = 2: No solutions.";

return;

}

if(n == 1) {

cout << "\n Result: 1";

return;

}

maxlen = Str(n\*n).length();

cout << "\n maxlen = " << maxlen;

if (n % 2 == 1) OMS(n);

else EMS(n);

}

main() {

for(int n = -3; n < 20; ++n) {

MS(n); Go();

}

// cout << Str(102030) << " " << Str(0);

cout << endl << "\n\n T h e E n d \n";

return 0;

}

### 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**

/\*

Name: MS.CPP

Copyright: (C) 2023

Author: Devcpp Fan

Date: 22-06-23 11:50

Description: Magic Square

\*/

#include <bits/stdc++.h>

using namespace std;

const int MN = 100;

int a[MN][MN];

int maxlen;

int N;

#define Num(i,j) (i)\*N+(j)+1

void Go() {

cout << " ? ";

fflush(stdin);

if (cin.get()=='.') exit(0);

}

string Str(int n) {

stringstream ss;

ss << n;

return ss.str();

}

void Print(int a[], int n, const char \* msg = "") {

cout << msg;

for (int i = 0; i < n; ++i) {

string s = Str(a[i]);

for(int j = s.length(); j <= maxlen; ++j) {

s = " " + s;

}

cout << " " << s;

}

}

void Print(int a[][MN], int n, const char \* msg = "") {

cout << msg;

for (int i = 0; i < n; ++i) {

Print(a[i], n, "\n ");

}

}

void Test(int n) {

int c = (n\*n+1)\*n/2;

int col, row;

int d1 = 0, d2 = 0;

int er = 0;

for(int i = 0; i < n; ++i) {

col = row = 0;

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

for(int j = 0; j < n; ++j) {

col += a[i][j];

row += a[j][i];

}

if(col != c) {

cout << "\n ERROR in row " << i;

++er;

}

if(row != c) {

cout << "\n ERROR in col " << i;

++er;

}

}

if(d1 != c) {

cout << "\n ERROR in diagonal 1";

++er;

}

if(d2 != c) {

cout << "\n ERROR in diagonal 2";

++er;

}

if(er == 0) cout << "\n CORRECT.";

}

void OMS(int n) {

for(int i = 0; i < n; ++i)

for(int j = 0; j < n; ++j)

a[i][j] = 0;

int i = 0, j = n/2;

a[i][j] = 1;

int nn = n\*n;

for(int k = 2; k <= nn; ++k) {

// (i,j) ?

--i; ++j;

if(i < 0 && j == n) {

// below (i,j)

i += 2; --j;

}

else if(i < 0) i = n-1;

else if(j == n) j = 0;

if(a[i][j] > 0) {

i += 2; --j;

}

a[i][j] = k;

}// for

Print(a,n,"\n Result: \n");

Test(n);

}

int OldNum(int i, int j) {

return N\*i + j + 1;

}

inline void Swap(int i, int j, int ii, int jj) {

a[i][j] = Num(ii,jj);

a[ii][jj] = Num(i,j);

}

// T: doi xung tam

// D: doi xung doc

// N: doi xung ngang

void EMS(int n) {

// ao xau mau

N = n;

int n2 = n/2;

int k = n2/2;

string s = "";

for(int i = 1; i <= k; ++i)

s += "T";

// neu n2 le: them DN

if(n2 % 2 == 1) s += "DN";

for(int i = s.length() + 1; i <= n2; ++i)

s += "#";

// Init a

for(int i = 0; i < n; ++i) {

for(int j = 0; j < n; ++j) {

a[i][j] = Num(i,j);

}

}

Print(a,n,"\n Init a: \n");

int n1 = n-1;

for(int i = 0; i < n2; ++i) {

// dien dong i

// cout << "\n Xau mau: " << s;

for(int j = 0; j < n2; ++j) {

switch(s[j]) {

case 'D': // doi xung doc (i,j) : (i, n-1-j)

Swap(i, j, i, n1-j); break;

case 'N': // doi xung ngang (i,j) : (n-q-i, j)n-1-j)

Swap(i, j, n1-i, j);

break;

case 'T': // doi xung tam

Swap(i, j, n1-i, n1-j);

Swap(i, n1-j, n1-i, j);

break;

} // switch

} // j

// quay xau mau

s = s[n2-1]+s.substr(0,n2-1);

} // i

Print(a,n,"\n Result a: \n");

Test(n);

}

void MS(int n) {

cout << "\n Magic Square of " << n;

if(n < 1 || n == 2) {

cout << "\n n <= 0 or n = 2: No solutions.";

return;

}

if(n == 1) {

cout << "\n Result: 1";

return;

}

maxlen = Str(n\*n).length();

cout << "\n maxlen = " << maxlen;

if (n % 2 == 1) OMS(n);

else EMS(n);

}

main() {

for(int n = 1; n < 20; ++n) {

MS(n); Go();

}

// cout << Str(102030) << " " << Str(0);

cout << endl << "\n\n T h e E n d \n";

return 0;

}