International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 2, Issue. 5, Sep.-Oct. 2012 pp-3583-3584 ISSN: 2249-6645

Magic Square

Dr. Mangala Gurjar
St. Xavier's College, Mumbai 400001

Abstract: *In this paper we will discuss an algorithm to generate a magic square of any odd order. We will also see computer programs for this algorithm in the programming languages C which is a popular programming language and in BASIC which is a higher level language suitable to translate algorithms. The programs can be run on computers to generate a magic square of order n x n for any odd value of n.*

Key words: *Algorithm, Array, BASIC Program, C-program, Magic square.*

I. Introduction

Magic squares have fascinated humans for many centuries. Following is a magic square created in 1514 by Albrecht Durer.

It looks like this.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

And what is the magic?
It is the number 34!

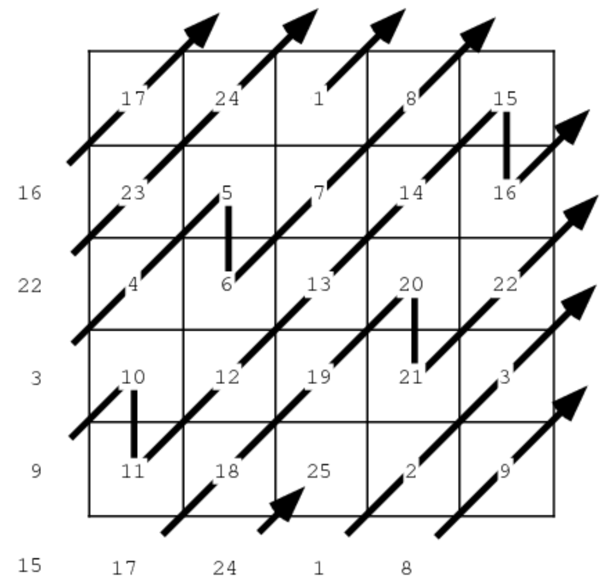
This number is the sum of the various fields within the magic square.

Since then many magic squares were made. The latest one is a 3125 x 3125 magic square made by Mr. Suresh Sutar of Kolhapur, Maharashtra in October 2011.

II. Algorithm

Following is an algorithm that can generate a magic square of size n x n for any odd value of n.
Step 1 Input an odd number n.
Step 2 Create an (n+1) x (n+1) integer array having all entries equal to zero.
Step 3 Set m equal to 1. Row number i to 1 and Column number j to (n+1)/2
Step 4 While m is not bigger than n^2 repeat the following steps up to step 11.
Step 5 Insert m in the (i,j)th place.
Step 6 Decrease i by 1 and increase j by 1. Increase m by 1.
Step 7 If i = 0 and j is not equal to n+1 then change i to n.
Step 8 If j =n+1 and i is not equal to zero then change j to 1.
Step 9 If i = 0 and j = n+1 then change i to 2 and j to n.
Step 10 If the (i,j)th entry is non zero then increase i by 2 and decrease j by 1.
Step 11 Increase m to m+1.
Step 12 Print the two dimensional array.

The schematic diagram for the above algorithm for n = 5 is given below.



III. C-Program

Following is the c implementation of this algorithm

```
# include "stdio.h"
# include "conio.h"
int main (int n)
{
printf("Enter\n");
scanf("%d" , &n);

int index = n;
int a[index+1][index+1];
int i,j,m;

for(i=1 ; i<=n ; i++)
          for(j=1 ; j<=n ; j++)
                    a[i][j]=0;

m = 1; i = 1; j = (n+1)/2 ;
While ( m <= n*n)
{
```

```
            a[i][j] = m ; m++ ; i = i - 1 ; j++ ;
            If (i ==0 && j<>n+1)then i = n ;
            If (i <> o && j == n+1) then j = 1;
            If(i == 0 && j == n+1) then {i = 2; j = n};
          If a[i][j]<>0 then {i = i+2 ; j = j-1}
}
for(i=1; i<=n ; i++)
            for(j=1 ; j<=n ; j++)
            {
            printf("%d", a[i][j]);
            if(j=n)    printf("\n");
            }
getch();
return 0;
}
```

IV. BASIC-Program

Following is the Basic program for the same algorithm.

```
10 REM Magic Square
20 INPUT "The dimension"; N
30 DIM A(N+1, N+1)
40 FOR I  = 1 To N+1
50 FOR J =  1 TO N+1
60 LET A(I, J) = 0
70 NEXT J
80 NEXT I
90 LET I = 1 : J = (N + 1)/2
100 FOR  M = 1 TO N*N
110 LET A(I, J) = M
120 LET I = I -1 : J = J + 1
130 IF I = 0 AND J <> N +1 THEN I = N
140 IF J = N + 1 AND I <> 0 THEN J = 1
150 IF I = 0 AND J = N + 1 THEN I = 2 : J = N
160 IF A(I, J)<> 0 THEN I = I + 2 : J = J -1
170 NEXT M
180 FOR I = 1 TO N
190 FOR J = 1 TO N
200 PRINT A(I, J) ;
210 NEXT J
220 PRINT
230 NEXT I
240 END
```

V. Conclusion

An n x n magic square for any odd large value of n can be formed using the above simple computer program. This eliminates claims made by lay people of having set a new record of making the largest magic square. Similar program can be written for even values of n also.

REFERENCES

[1] R.G. Dromey, How to Solve it by computers Prentice-Hall India.

[2] Kenneth H. Rosen : Discrete Mathematics and Its Applications, McGraw Hill