**Magic Cube Algorithm:**

|  |  |
| --- | --- |
| Number | Co-ordinate |
| 1 | (1,1,0) |
| 2 | (0,1,2) |
| 3 | (2,1,1) |
| 4 | (1,0,2) |
| 5 | (0,0,1) |
| 6 | (2,0,0) |
| 7 | (1,2,1) |
| 8 | (0,2,0) |
| 9 | (2,2,2) |
| 10 | (2,2,0) |
| 11 | (1,2,2) |
| 12 | (0,2,1) |
| 13 | (2,1,2) |
| 14 | (1,1,1) |
| 15 | (0,1,0) |
| 16 | (2,0,1) |
| 17 | (1,0,0) |
| 18 | (0,0,2) |
| 19 | (0,0,0) |
| 20 | (2,0,2) |
| 21 | (1,0,1) |
| 22 | (0,2,2) |
| 23 | (2,2,1) |
| 24 | (1,2,0) |
| 25 | (0,1,1) |
| 26 | (2,1,0) |
| 27 | (1,1,2) |

If you observe the above table all the co-ordinates of the numbers form a pattern which appears like this

The First element is stored at location (n/2,n/2,0)

To get the next element location

1. subtract the x co-ordinate by 1
2. subtract the y co-ordinate only when the number if divided by n leaves remainder as 1
3. subtract the z co-ordinate by 1 except in the case where if the number when divided by n leaves the remainder as 1 then you add the z co-ordinate by 1.

In any of the above cases if you get any co-ordinate value as -1 then replace it by n-1 and if you get any co-ordinate as n then replace it with 0.

After completing the above steps if you get x and y co-ordinate as equal and z as 0 then increase the x and y co-ordinate by 1.

After doing the above step if you get x and y co-ordinate as n and z as 0 then replace x and y co-ordinate with 0.

**Functions used for tic tac toe algorithm:**

printboard()

Prints the Tic Tac Toe Board

hscore():

returns number of legal lines that human scored

cscore():

returns number of legal lines that human scored

collinear(int a, int b, int c)

Determines if three points are collinear or not

legalline(int x, int y, int z) function:

Checks if three points x, y, z are collinear and sum to 42.

cpossiblewin() function:

If it returns 0 then there is no possible win for the computer, if the functions returns a value it means that by marking that cell computer can score the line.

hpossiblewin() function:

If it returns 0 then there is no possible win for human , if the functions returns a value it means that by marking that cell human can score the line so the computer blocks the line.

Rank is defined as sum of the no.of lines passing through the given cell which does not have human cell in it and the no.of lines passing through given cell which only have computer cell in it

Ex: Computer marked 14 and human marked 10

Rank of 8 = 4 = 1(8,12,22 line) + 1(8,15,19 line)+0(because 8,24,10 has been already marked by human)+ 1(8,14,17 line)+1(8,14,17 line)(because 14 is already marked by computer we have advantage in marking 8)

rank(int n) function:

returns the rank of the cell having value n

hrank(int n) function:

returns the rank of the cell having value n from human perspective.

computer\_choice() function internally uses rank and hrank function which determine the ranks of the cells from both computer and human perspective and returns cell value having the highest rank.

* If human has more advantage in marking the cell than computer marking some other cell, the computer blocks that cell so that human can’t take advantage. This stops human from having double wins or triple wins.
* If computer has more advantage in marking a cell then computer marks the cell and gains advantage. This helps computer to achieve double wins or triple wins.

**Tic-Tac-Toe Algorithm:**

Tic tac toe algorithm that we used is divided into three steps:

1.Checking if computer has possible win with help of “cpossiblewin()”, If it’s possible to win computer goes for it.

2. Checking if human has possible win with help of “hpossiblewin()”, If it’s possible to win computer blocks it.

3.If the above both cases fail the computer marks the most suitable cell using “computer\_choice()” which returns suitable cell number for computer to perform the next move.