Probably most difficult part is to loop in m3x in vertical

for (int row = 7; row >= 0; row--)

{

for (int col = 7; col >= 0; col--)//was col >= 1 becouse miss understood looping in vertical

{

if (arr[row, col] == 0)

{// find empty slot and save coordinates

isEmpty = true;

emptyCol = col;

emptyRow = row;

//vertical search and replace

for (int searchCol = emptyRow - 1; searchCol >= 0; searchCol--)

{//-1 to start from next position donw -> up direction

if (arr[searchCol, col] == 1)//iterate over rows in order to serch in specific collumn col index is dif than column itself

{//find occupy slot

arr[emptyRow, col] = 1;

emptyRow--;//go upper next in colonm

arr[searchCol, col] = 0;

}

}

}

}

}

* We start looping through the m3x as it is on the scheme, if we loop through other organized m3x we will change it this m3x is bitwise based

int[,] arr = new int[8, 8];

for (int row = 0; row < 8; row++)

{

//standart way to declare arr of bits from number

int number = int.Parse(Console.ReadLine());

for (int col = 0; col < 8; col++)

{

arr[row, col] = (number >> col) & 1; // 1&1= 1 ;

}

}

* Character based :

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

{

numbers[i] = int.Parse(Console.ReadLine());

line = Convert.ToString(numbers[i], 2).PadLeft(n, '0').ToCharArray();

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

{

grid[i, j] = line[j];

}

}

And it’s loop:

for (int row = n - 1; row >= 1; row--)// access last el in rowFor

{

for (int col = n - 1; col >= 0; col--)

{

strartRow = row;

if (grid[row, col] == '0')// current element

{

for (int rowFor = row - 1; rowFor >= 0; rowFor--)

{

if (grid[rowFor, col] == '0')

{

continue;

}

else if (grid[rowFor, col] == '1')

{

grid[strartRow, col] = '1';

grid[rowFor, col] = '0';// clear if found

strartRow--;

// PrintM3x(n, grid);

}

}

}

else if (grid[row, col] == '1')

{

continue;

}

}

}

So back to basics **looping in vertical** means you **station the column index** and **loop up** word through the **row index(vertical)**  
 vise-versa is true station the row index and loop through the column (horizontal)

for (int searchCol = emptyRow - 1; searchCol >= 0; searchCol--)

{//-1 to start from next position donw -> up direction

if (arr[searchCol, col] == 1)//iterate over rows in order to serch in specific collumn col index is dif than column itself

{//find occupy slot

arr[emptyRow, col] = 1;

emptyRow--;//go upper next in colonm

arr[searchCol, col] = 0;

}

}