Report

Problem: Have to write a program that can take a matrix as input and check if that matrix is a magic square or not.

Solution:

Basic idea about how to solve the problem:

So, to solve this problem we have to go forward step by step.

First, we have to worry about the input. We have to write the program in such a way that it can take a matrix as input. So, we have to use 2D array for the input. Otherwise we can't take matrix as the input.

The second thing we need to worry about is the condition we need to check for the matrix to be a magic square.
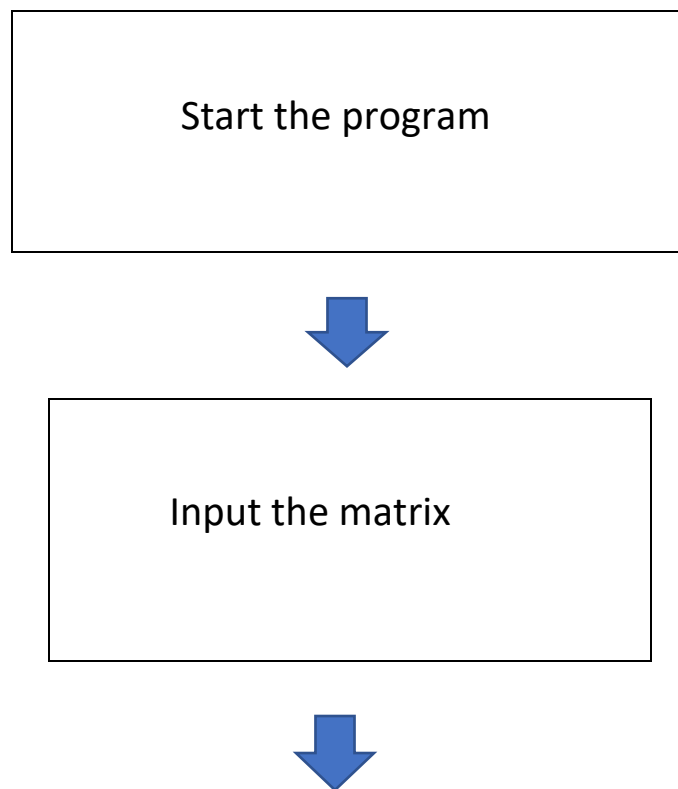
The condition is that sum of any row, column or diagonal of a magic square is equal to the highest value of the number of that matrix.

So, to check this condition, first we need to check if sums of both diagonals is same. If not then it is not a magic square. If it is, then we

move onto the next part. Then we need to check the sum of all the columns, same rule applies here as well. For the last part we are going to check if sum of all the rows are same. Again, the same rule goes here.

If the matrix meets all three conditions then it's a magic square. Other wise it's not magic square.

Flow chart:

```
┌─────────────────────────────────┐
│                                 │
│        Start the program        │
│                                 │
└─────────────────────────────────┘
                 ↓
┌─────────────────────────────────┐
│                                 │
│         Input the matrix        │
│                                 │
└─────────────────────────────────┘
                 ↓
```

```
┌─────────────────────────────┐
│   Check sums of all the     │
│         diagonals           │
└─────────────────────────────┘
              ↓

      Are all the sums equal?

              ↓
    ←─────────────────────→
┌──────────┐              ┌──────────┐
│   Yes    │              │   No     │
└──────────┘              └──────────┘
                               ↓
                    ┌─────────────────────────┐
                    │ Print "It's not a Magic │
                    │ Square"                 │
                    │                         │
                    └─────────────────────────┘

    ┌─────────────────────────────┐
    │  Find the sum of columns    │
    │                             │
    │                             │
    └─────────────────────────────┘
                  ↓
```
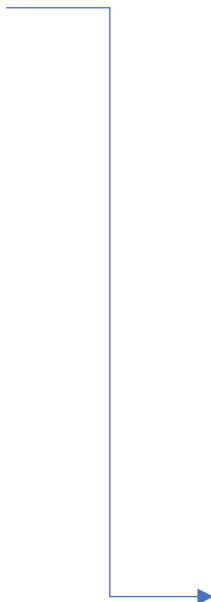
Find the sum of rows
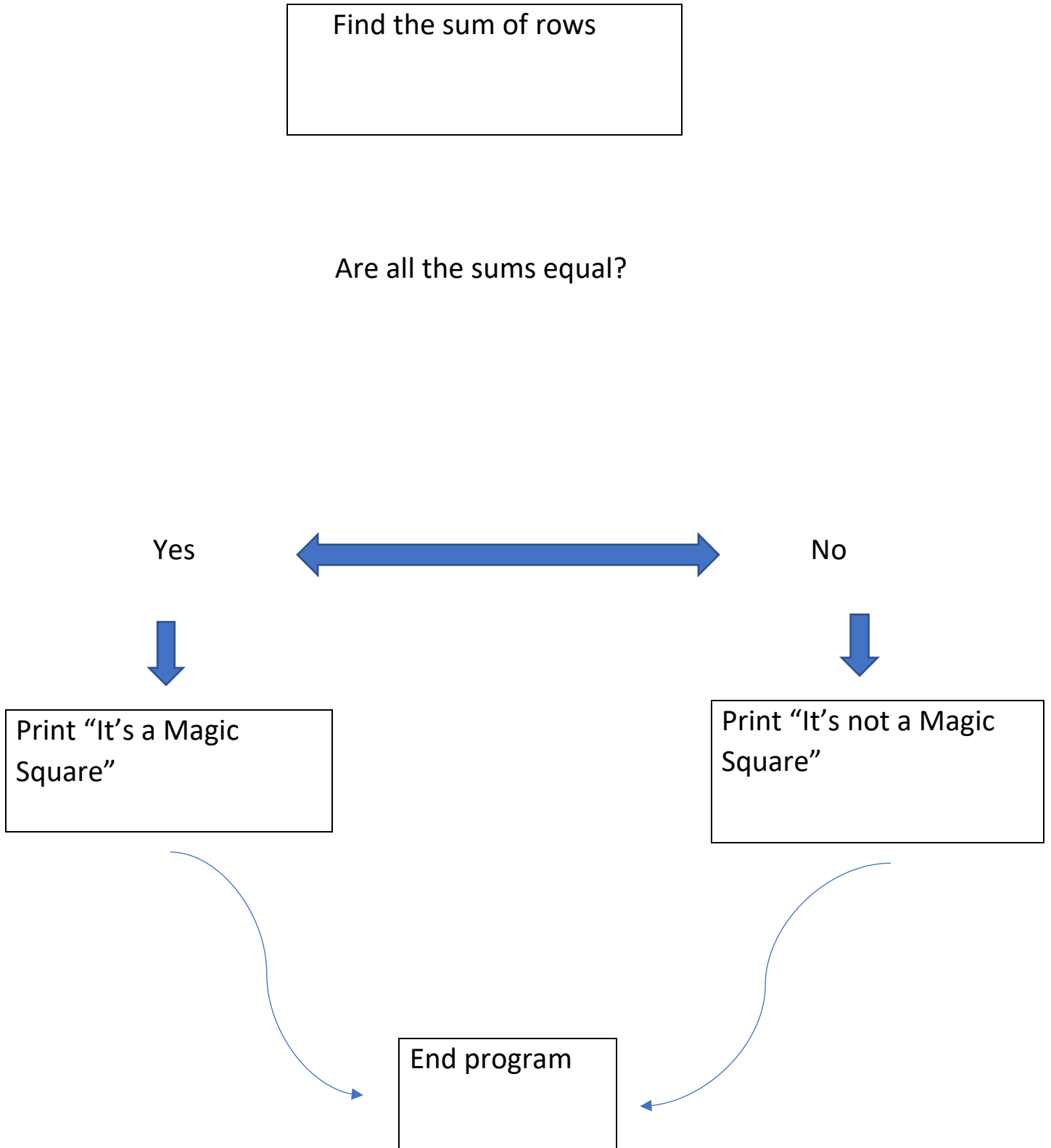
Are all the sums equal?

Yes ⟷ No

Print "It's a Magic Square"

Print "It's not a Magic Square"

End program

Explanation:

```c
// int y [3][3],i,j,sumd1=0,sumd2=0,f,sumr=0,sumc=0;



printf ("Enter matrix elements..\n");


for (i=0;i<3; i++) {


   for (j=0; j<3;j++){
      scanf("%d", y [i][j]); //
```

In the first line of this portion of the program, it declared the type and name of the variables.

The last 3 lines is written so , the user can input the matrix

```
//for (i=0;i<3; i++) {

   for (j=0; j<3;j++){
      printf("3%d", y [i][j]);
} printf ("\n");
 } //
```

These three lines is written, so the user can see the matrix he/she just entered.

```
//for (i=0;i<3; i++) {

   for (j=0; j<3;j++){
      if (i==j) {

         sumd1=sumd1+y[i][j ]; // sum of  main diagonal
```

```
                }


        if (i+j==3-1)

        {

            sumd2=sumd2+y[i][j]; //sum of second diagonal

        } }
} //
```

So, now the program will find out the sums of the 2 diagonals.

```
// if (sumd1!=sumd2){

        f=1; // we are gonna use this kind of like a switch

    }

    else{

        for (i=0;i<3; i++) {


            for (j=0; j<3;j++){

                sumc=sumc+y[j][i];
```

```
        sumr=sumr+y[i][j];
} }//
```

Now the program will find out if the sums of the 2 diagonals are equal. If they are not equal we are going to put a value in f. f is going to work like a switch in this program. We are going to put only 2 values in it 0 and 1, depending on the condition.

If the sums are equal then it's going to find out the sums of both columns and rows.

```
  if (sumc!=sumd1){
    f=1;
  }


  else if (sumr!=sumd1){


    f=1;}
  else {
```

```
f=0;  } } //
```

Now the program will check if the sum of rows and columns are equal to the sum of the diagonal.

If they are we are gonna put 0 in f. Otherwise we are going to put 1 in f.

```
// if (f==0) {

    printf ("\n It's a square matrix\n");

  }
else if (f==1){

  printf("\n It's not a square matrix\n");
} } //
```
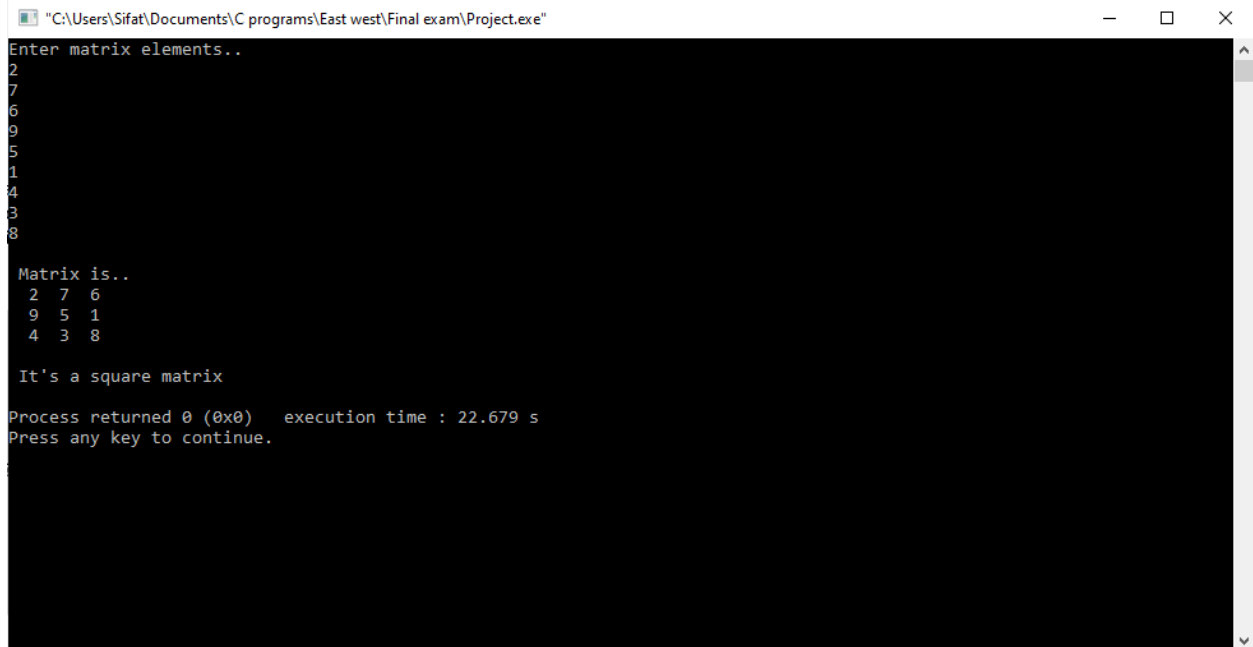
In these lines we are finally going to check the value of f, the switch of the program.

If the value of f is equal to 0 then, the program is going print that the matrix was a square matrix. Otherwise if f=1 then, the program is going to print that the matrix is not a square matrix
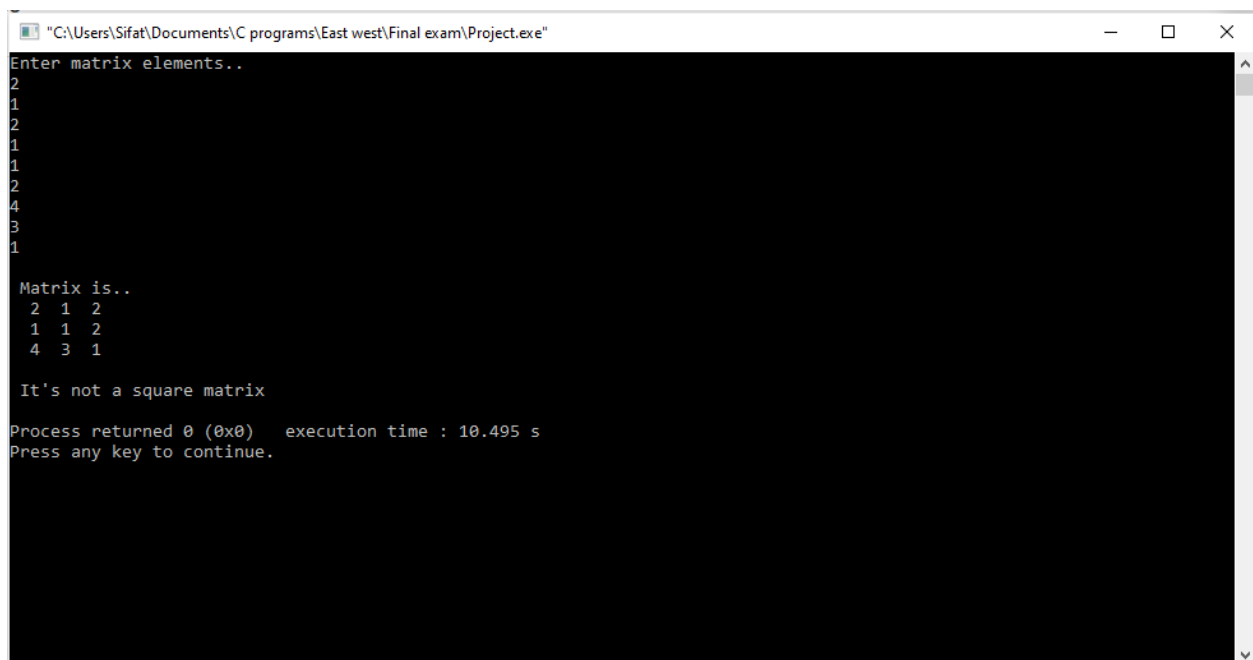
Output:


```
"C:\Users\Sifat\Documents\C programs\East west\Final exam\Project.exe"               —   □   ×
Enter matrix elements..
2
7
6
9
5
1
4
3
8

 Matrix is..
  2  7  6
  9  5  1
  4  3  8

 It's a square matrix

Process returned 0 (0x0)   execution time : 22.679 s
Press any key to continue.
```

Example of square matrix


```
"C:\Users\Sifat\Documents\C programs\East west\Final exam\Project.exe"               —   □   ×
Enter matrix elements..
2
1
2
1
1
2
4
3
1

 Matrix is..
  2  1  2
  1  1  2
  4  3  1

 It's not a square matrix

Process returned 0 (0x0)   execution time : 10.495 s
Press any key to continue.
```

Example of non-square matrix