```
//Name-Tasfique Enam

//Student ID- 5886429

//Assignment 1

#include <iostream>

#include <fstream>

#include <string>

#include <bits/stdc++.h>

#include <iomanip>

#include "extention.h"


using namespace std;


//function prototype



int main () {

  int option;

  int n;

  int input;

  string fileName;


  int** magicSquareArray;

  fstream ifs;

  bool check = false;

  bool check2=false;




  do { //do loop for menu function.

    cout << endl;

    cout << "Options-----------------" << endl;
cout << "1- Construct Magic Square" << endl;

cout << "2- Check if it is Magic Square or not" << endl;

cout << "3- Generate Second Magic Square" << endl;

cout << "0- Quit" << endl;


cout << "\nEnter option: ";

cin >> option;



switch(option) { //switch statement


  case 1: { //case 1 from the menu

    cout << "Enter an odd number ";

    cin >> setw(5) >> n;

    while(n!=3 && n!=5 && n!=7 && n!=9) { //while loop for validation.

      cout << "Invalid input please, try again. "<< endl;

      cout << "Enter an odd number ";

      cin >> setw(5) >> n;

    }

    magicSquareArray = new int *[n];


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

      magicSquareArray[i] = new int [n];

    }

    cout << setw(5) << n << endl;

    cout << " " << endl;

    constructMagicSquare(n, magicSquareArray);


    cout << "Do you want to Save this Magic Square? Enter 1 for Yes or 0 for No" << endl;

    cin >> input;

    while(input!=0 && input!=1) {
```

```
   cout << "Invalid Input, Try again. " << endl;
   cout << "Do you want to Save this Magic Square? Enter 1 for Yes or 0 for No" << endl;
   cin >> input;
}
if(input == 1) { //if option1 was selected.
   cout << "Enter the file Name " << endl;
   cin >> fileName;

   fileName = fileName + ".txt";
   ifs.open(fileName.c_str(), ios::out);
   if (ifs.is_open()) {
      ifs << n;
      for(int i=0; i<n; i++) {
            ifs << endl;
         for(int j=0; j<n; j++) {
            ifs<<magicSquareArray[i][j]<<" ";
         }
      }
      cout << "Successfully written. " << endl;
      ifs.close();
   } else {
      cout << "Unable to open the file." << endl;
   }
} else if(input == 0){
   cout << "File haven't been written" << endl;
}
for(int i = 0; i < n; ++i) {
   delete[] magicSquareArray[i];
}
//Free the array of pointers
delete[] magicSquareArray;
```

```
    check = false;
    check2 =true;

} break;

case 2: {

        if(check2==true)
        {
            magicSquareArray = new int *[n];

        for(int i=0; i<n; ++i) {
            magicSquareArray[i] = new int [n];
        }
        cout << "Enter the name of the TXT File you want to open. " << endl;
        cin >> fileName;
        fileName = fileName + ".txt";

        ifs.open(fileName.c_str(), ios::in);
        if(!ifs) {
            cout << "File could not be opened. " << endl;
        } else {
            ifs >> n;
            for(int i=0; i<n; i++) {
                    //ifs >> endl;
                for (int j=0; j<n; j++) {
                    ifs >> magicSquareArray[i][j]; ///do i need to create an array for this or not?
                }
            }
            if(isMagicSquare(n, magicSquareArray)) {
                cout << "It is a Magic Square" << endl;
                check=true;
            } else {
                cout << "It is not a Magic Square" << endl;


            }
            ifs.close();
        }
        for(int i = 0; i < n; ++i) {
            delete[] magicSquareArray[i];
        }
        }
        else{
            cout << "Please, create a Magic Square first!" << endl;
        }
        //Free the array of pointers
        delete[] magicSquareArray;



        }
         break;


case 3: {


    if(check == true) {
        magicSquareArray = new int *[n];


        for(int i=0; i<n; ++i) {
            magicSquareArray[i] = new int [n];
        }


        fileName = fileName + ".txt";
```

```
      if(!ifs) {
         cout << "File could not be opened. " << endl;
      } else {
         constructMagicSquare(n, magicSquareArray);
         ifs >> n;
         for(int i=0; i<n; i++) {
            for (int j=0; j<n; j++) {
              ifs >> magicSquareArray[i][j];
            }
         }
         cout << " \n" << endl;
         createAnotherMagicSquare(n, magicSquareArray);
         ifs.close();
      }
      //createAnotherMagicSquare(n, magicSquareArray);
       for(int i = 0; i < n; ++i) {
         delete[] magicSquareArray[i];
      }
      //Free the array of pointers
      delete[] magicSquareArray;

   } else if (check == false) {
      cout << "Your TXT File doesn't have a Magic Square " << endl;
   }

} break;

case 0: {
   cout << "Thanks for using the Program " << endl;
   exit(0);
}
```

```
        }

    }while(option>=0 && option<=3);

}



//Name-Tasfique Enam
//Student ID- 5886429
//Function Class

#include <iostream>
#include <fstream>
#include <string>
#include <bits/stdc++.h>
#include <iomanip>
#include "extention.h"

using namespace std;

//for creating magic square.
void constructMagicSquare (int n, int** magicSquareArray) {
    int squareSize = n*n;
    int column = n/2;
            int row = 0;

            for(int i=0; i<n; i++) {
        magicSquareArray[i] = new int [n];

            }
    for(int i=1; i<=squareSize; i++) {
        magicSquareArray[row][column] = i;

        row--;
        column++;

        if(i%n == 0) {
            row += 2;
            --column;
        } else {
            if (column==n)
                column -= n;
            else if (row<0)
                row += n;
        }
    }

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

        for(int j=0; j<n; j++) {
            cout<<setw(5)<< magicSquareArray[i][j] << " ";
        }

        cout<< endl;
    }


}

//for validating magic square.
bool isMagicSquare(int n, int** magicSquareArray) {
```

```
int sum =0;
int sum2=0;

//prime diagonal
for (int i=0; i<n; i++) {
   sum = sum + magicSquareArray[i][i];
}
//secondary check
for (int i = 0; i < n; i++)
   sum2 = sum2 + magicSquareArray[i][n-1-i];

if(sum!=sum2)
   return false;

for (int i = 0; i < n; i++) {
   int rowSum = 0;
   for (int j = 0; j < n; j++)
      rowSum += magicSquareArray[i][j];

   // check if every row sum is
   // equal to prime diagonal sum
   if (rowSum != sum)
      return false;

   // For sums of Columns
   for (int i = 0; i < n; i++) {
         int colSum = 0;
      for (int j = 0; j < n; j++)
         colSum += magicSquareArray[j][i];

      // check if every column sum is
      // equal to prime diagonal sum
      if (sum != colSum)
      return false;


    }
    return true;


  }


}
```

/*To create another magic square i have switched the last row of data with the first row of data. To achieve this,

I have created a nested for loop the first for loop goes through the column, and the second one goes through the row,

given that the array index location is the same for the columns, there is no point of changing the array index of those given data,

and for the row array index, i have decremented the value by minus 1, [n-1-i] and the values got switched for the first row and column.

int b, temporarily takes the data of magicSquareArray, and the MagicSquareArray takes the new data with the changed row in the for loop.

$$\begin{array}{cccc} & [0] & [1] & [2] \\ [0] & 8 & 1 & 6 \\ [1] & 3 & 5 & 7 \\ [2] & 4 & 9 & 2 \end{array}$$

In this representation of the matrix, the last row value have been switched with the first row of data, with the explained method above.

```
    [0]  [1] [2]

[0]   4   9   2

[1]   3   5   7

[2]   8   1   6



*/


//for creating another magic square.

void createAnotherMagicSquare (int n, int** magicSquareArray) {

   int temporary;

   for (int i=0; i<n/2; i++) {

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

        temporary = magicSquareArray[i][j];

        magicSquareArray[i][j] = magicSquareArray[n-1-i][j];

        magicSquareArray[n-1-i][j] = temporary;

    }

   }


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

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

        cout <<setw(5)<< magicSquareArray[i][j] << " ";

    }

    cout << endl;

   }


}
```

```
/*void createAnotherMagicSquare (int n, int** magicSquareArray) {


    for(int i=n-1; i>=0; i--) {

        cout << endl;

            for (int j=n-1; j>=0; j--) {

                cout << magicSquareArray[i][j] << " ";

            }

    }

    cout<< endl;


}*/


/*void createAnotherMagicSquare (int n, int** magicSquareArray) {

    int row = 0;

    int col = n/2;


    for (int i=2; i<=n*n; i++) {

        int newColumn = ((col - 1)+n) % n;

        int newRow = ((row - 1)+n) % n;

        if (magicSquareArray[newRow][newColumn] == 0) {

            row = newRow;

            col = newColumn;

        } else {

            row = (row + 1) % n;

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

                cout << magicSquareArray[i][j] << " ";

            }
    }
  }

}*/
//Name-Tasfique Enam
//Student ID- 5886429
//Header file
#ifndef EXTENTION_H
#define EXTENTION_H
void constructMagicSquare (int n, int** magicSquareArray);
bool isMagicSquare(int n, int** magicSquareArray);
void createAnotherMagicSquare (int n, int** magicSquareArray);
#endif // EXTENTION_H
```