#include<iostream>

#include<stdio.h>

#include<math.h>

using namespace std;

int alpha(int x) // function to be used to tell whether the matrix is allowed or not

{

if (x == 1)

{

cout << "This matrix is not allowed" << endl;

}

else

{

cout << "This matrix is allowed" << endl;

}

return x;

}

int main()

{

int k;

int n;

cout << "Enter the size of square matrix i.e. n:" << endl; //the matrix is of size n/n

cin >> n;

int i, j;

int b;

char x, o;

cout << "Enter which characters do you want" << endl;

cin >> x>>o;

char\*\* A = new char\* [n]; //declaring an array/matrix which has the size given by the user

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

{

A[i] = new char[n];

}

for (i = 0; i < n; i++) //intitiallizing the array to o

{

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

{

A[i][j] = o;

}

}

for (i = 0; i < n; i++) //to take in x and o for each element position

{

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

{

f1:

cout << "Enter the element for [" << i << "]th and [" << j << "]th element" << endl;

cin >> A[i][j];

if (A[i][j] != x && A[i][j] != o)

{

cout << "ReEnter Values :" << endl;//code for removing all possible human errors

goto f1;

}

}

}

for (i = 0; i < n; i++) //just to show how the matrix looks like

{

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

{

cout << "\t" << A[i][j];

}

cout << "\n";

}

//calculation part

for (i = 2; i <= n; i++) //i reprents which internal matrix is it solving... like it starts by solving

//internal 2X2 matrix. The it solves 3X3 matrix...and so on

{

for (j = 0; j <= n - i; j++)//j reprents the number of the row

{

for (k = 0; k <= n - i; k++)//k represents the number of column

{

if (A[j][k] == x && A[j + i - 1][k] == x && A[j + i - 1][k + i - 1] == x && A[j][k + i - 1] == x)

{

b = 1;

goto f2;//this statement help to not go into furthur calculation

break;

}

else

{

b = 0;

}

}

}

}

//same code repeated for o

for (i = 2; i <= n; i++)

{

for (j = 0; j <= n - i; j++)

{

for (k = 0; k <= n - i; k++)

{

if (A[j][k] == o && A[j + i - 1][k] == o && A[j + i - 1][k + i - 1] == o && A[j][k + i - 1] == o)

{

b = 1;

goto f2;

break;

}

else

{

b = 0;

}

}

}

}

f2://here f2 directs us to the above code

alpha(b);

return 0;

}