Aya Hesham 230489 Assignment 2 CS102 Group A : Q 1, 4, 5

1-)

X-O

Write a program to make 2 player of (X-O) game to play.

The board is just (3x3).

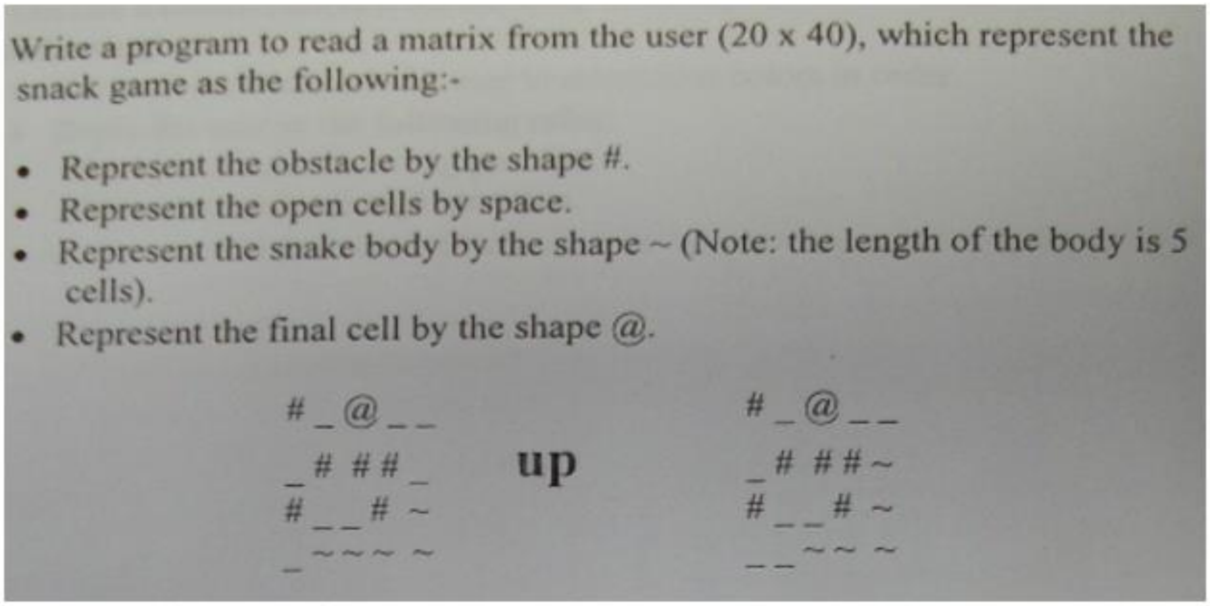
Each player should select cell , and your program will fill it by (X) or (O) based in the player turn.

```
#include <iostream>
using namespace std;

void Check(char x[][3], char p1, char p2, int& check)
{
	check = 0;
	for (int i = 0; i < 3; i++)
	{
	   if (x[i][0] == x[i][1] && x[i][1] == x[i][2])
	    {
			check = 1;
		if (x[i][0] == p1) { cout << "Congratulations!! Winner is Player 1"; }
			else { cout << "Congratulations!!Winner is Player 2"; }
			break;
		}
		if (x[0][i] == x[1][i] && x[1][i] == x[2][i])
		{
			check = 1;
		 if (x[i][0] == p1) { cout << "Congratulations!! Winner is Player 1"; }
			else { cout << "Congratulations!!Winner is Player 2"; }
			break;
		}
	}
	if (x[0][0] == x[1][1] && x[1][1] == x[2][2])
	{
		check = 1;
		if (x[0][0] == p1) { cout << "Congratulations!! Winner is Player 1"; }
		else { cout << "Congratulations!!Winner is Player 2"; }

	}
	if (x[0][2] == x[1][1] && x[1][1] == x[2][0])
	{
		check = 1;
		if (x[0][2] == p1) { cout << "Congratulations!! Winner is Player 1"; }
		else { cout << "Congratulations!!Winner is Player 2"; }

	}

}

void main()
{
	char x[3][3], p1, p2;
	int sr, sc, check = 0;
```

```
x[0][0] = '1', x[0][1] = '2', x[0][2] = '3', x[1][0] = '4', x[1][1] = '5',
 x[1][2] = '6', x[2][0] = '7', x[2][1] = '8', x[2][2] = '9';

	for (int r = 0; r < 3; r++)
	{
		for (int c = 0; c < 3; c++)
		{
			cout << x[r][c] << " ";
		}
		cout << endl;
	}

	cout << "Player 1 choose X or O" << endl;
	cin >> p1;
	if (p1 == 'X')
	{
		p2 = 'O';
	}
	else
	{
		p2 = 'O';
	}

	for (int i = 0; ; i++)
	{
		if (i % 2 == 0)
		{
			cout << "Player 1 turn" << endl;
			cout << endl;
			cout << "Enter row" << endl;
			cin >> sr;
			cout << "Enter column" << endl;
			cin >> sc;

			x[sr][sc] = p1;
			Check(x, p1, p2, check);
			if (check == 1)
			{
				break;
			}

		}
		if (i % 2 != 0)
		{
			cout << "Player 2 turn" << endl;
			cout << endl;
			cout << "Enter row" << endl;
			cin >> sr;
			cout << "Enter column" << endl;
			cin >> sc;

			x[sr][sc] = p2;
			Check(x, p1, p2, check);
			if (check == 1)
			{
				break;
			}
```

```
            }

            for (int r = 0; r < 3; r++)
            {
                  for (int c = 0; c < 3; c++)
                  {
                        cout << x[r][c] << " ";
                  }
                  cout << endl;
            }

            int ct = 0;
            for (int r = 0; r < 3; r++)
            {
              for (int c = 0; c < 3; c++)
              {
                if (x[r][c] != '1' && x[r][c] != '2' && x[r][c] != '3' &&
                    x[r][c] != '4' && x[r][c] != '5' && x[r][c] != '6' &&
                     x[r][c] != '7' && x[r][c] != '8' && x[r][c] != '9')
                        {
                              ct++;

                        }
                  }

            }
            if (ct == 9)
            {
                  cout << "It's a tie :(" << endl;
                  break;
            }
      }

      cout << endl;
      for (int r = 0; r < 3; r++)
      {
            for (int c = 0; c < 3; c++)
            {
                  cout << x[r][c] << " ";
            }
            cout << endl;
      }
}
```

4-)



Write a program to read a matrix from the user (20 x 40), which represent the snack game as the following:-

- Represent the obstacle by the shape #.
- Represent the open cells by space.
- Represent the snake body by the shape ~ (Note: the length of the body is 5 cells).
- Represent the final cell by the shape @.

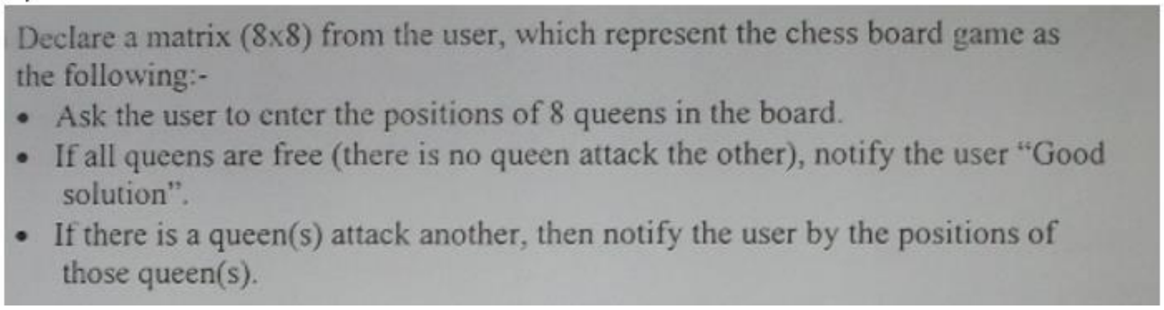
```
# _ @ _ _                  # _ @ _ _
_ # ## _        up         _ # ## ~
# _ _ # ~                  # _ _ # ~
_ ~~~ ~                    _ _ ~~ ~
```

```
#include <iostream>
using namespace std;
void main()
{
	char x[20][40];
	int s[5][2];
	char uc;

	cout << "Enter matrix as follows:" << endl;
	cout << endl;
	cout << "Represent obstacle by #" << endl << "Represent open area by space" 
<< endl << "Represent final cell by @" << endl;
	cout << endl;

	for (int r = 0; r < 20; r++)
	{
		for (int c = 0; c < 40; c++)
		{
			cin>> x[r][c];
		}
	}

	cout << endl;
	cout << "Enter snake (5 cells represented as ~)";
	for (int r = 0; r < 5; r++)
	{
		for (int c = 0; c < 2; c++)
		{
			cin>> s[r][c];
		}
	}

	int xr, xc;
```

```
        for (int r = 0; r < 5; r++)
        {
                int c = 0;
                xr = s[r][c];
                xc = s[r][c + 1];
                x[xr][xc] = '~';

        }
        cout << endl;
        for (int r = 0; r < 20; r++)
        {
                for (int c = 0; c < 40; c++)
                {
                        cout << x[r][c] << " ";
                }
                cout << endl;
        }

        int check = 0;
        for (;check!=1 ;)
        {
          cout << "Enter direction ( u for up , d for down , r for right, l for left"
<< endl;
                cin >> uc;


                if (uc == 'u')
                {
                        x[s[4][0]][s[4][1]] = '.';
                        for (int i = 4; i > 0; i--)
                        {
                                s[i][0] = s[i - 1][0];
                                s[i][1] = s[i - 1][1];
                        }

                        s[0][0] = s[0][0] - 1;

                        for (int r = 0; r < 5; r++)
                        {
                                int c = 0;
                                xr = s[r][c];
                                xc = s[r][c + 1];

                                if (x[xr][xc] == '@')
                                {
                                        cout << endl;
                                        cout << "WINNER!!" << endl;
                                        check = 1;
                                        break;
                                }
                                if (x[xr][xc] == '#')
                                {
                                        cout << endl;
                                        cout << "YOU LOST" << endl;
                                        check = 1;
                                        break;
                                }
                x[xr][xc] = '~';

            }
        }

        if (uc == 'd')
        {
            x[s[4][0]][s[4][1]] = '.';
            for (int i = 4; i > 0; i--)
            {
                s[i][0] = s[i - 1][0];
                s[i][1] = s[i - 1][1];
            }

            s[0][0] = s[0][0] + 1;

            for (int r = 0; r < 5; r++)
            {
                int c = 0;
                xr = s[r][c];
                xc = s[r][c + 1];

                if (x[xr][xc] == '@')
                {
                    cout << endl;
                    cout << "WINNER!!" << endl;
                    check = 1;
                    break;
                }
                if (x[xr][xc] == '#')
                {
                    cout << endl;
                    cout << "YOU LOST" << endl;
                    check = 1;
                    break;
                }

                x[xr][xc] = '~';

            }
        }

        if (uc == 'l')
        {
            x[s[4][0]][s[4][1]] = '.';
            for (int i = 4; i > 0; i--)
            {
                s[i][0] = s[i - 1][0];
                s[i][1] = s[i - 1][1];
            }

            s[0][1] = s[0][1] - 1;

            for (int r = 0; r < 5; r++)
            {
                int c = 0;
                xr = s[r][c];
                xc = s[r][c + 1];

                if (x[xr][xc] == '@')
                {
                        cout << endl;
                        cout << "WINNER!!" << endl;
                        check = 1;
                        break;
                }
                if (x[xr][xc] == '#')
                {
                        cout << endl;
                        cout << "YOU LOST" << endl;
                        check = 1;
                        break;
                }

                x[xr][xc] = '~';

        }
}

if (uc == 'r')
{
        x[s[4][0]][s[4][1]] = '.';
        for (int i = 4; i > 0; i--)
        {
                s[i][0] = s[i - 1][0];
                s[i][1] = s[i - 1][1];
        }

        s[0][1] = s[0][1] + 1;

        for (int r = 0; r < 5; r++)
        {
                int c = 0;
                xr = s[r][c];
                xc = s[r][c + 1];

                if (x[xr][xc] == '@')
                {
                        cout << endl;
                        cout << "WINNER!!" << endl;
                        check = 1;
                        break;
                }



                if (x[xr][xc] == '#')
                {
                        cout << endl;
                        cout << "YOU LOST" << endl;
                        check = 1;
                        break;
                }
```

```
                    x[xr][xc] = '~';

                }
            }

            cout << endl;
            for (int r = 0; r < 20; r++)
            {
                for (int c = 0; c < 40; c++)
                {
                    cout << x[r][c] << " ";
                }
                cout << endl;
            }
        }
}
```

5-)



Declare a matrix (8x8) from the user, which represent the chess board game as the following:-

- Ask the user to enter the positions of 8 queens in the board.
- If all queens are free (there is no queen attack the other), notify the user "Good solution".
- If there is a queen(s) attack another, then notify the user by the positions of those queen(s).

```
#include <iostream>
using namespace std;
void Check(char x[][8], int r, int c, int& check)
{
        int ct = 0;
        //row
        for (int i = 0; i < 8; i++)
        {
                if (x[r][i] == '&')
                {
                        ct++;
                        if (ct > 1)
                        {
                                check = 1;
                        }
                }
        }

         ct = 0;
        //column
        for (int i = 0; i < 8; i++)
        {
                if (x[i][c] == '&')
                {
                        ct++;
                        if (ct > 1)
                        {
                                check = 1;
                        }
                }
        }
        //DiagonalUpLeft
        ct = 0;
        for (int i = 0; i < 8; i++)
        {
                if (x[i][i] == '&')
                {
                        ct++;
                        if (ct > 1)
                        {
                                check = 1;
                        }
                }
        }
        //DiagonalDownLeft
        ct = 0;
        int k = 0;
        for (int i = 7; i > 0 && k <8; i--,k++)
        {
                if (x[i][k] == '&')
                {
                        ct++;
                        if (ct > 1)
                        {
                                check = 1;
                        }
                }
        }

}
void main()
{
        char x[8][8];
        int r=0, c=0,check=0;

        cout << "Enter positions of queens" << endl;
        for (int i = 0; i < 8; i++)
        {
                cout << "Queen " << i + 1 << endl;
                cout << "Enter row" << endl;
                cin >> r;
                cout << "Enter column" << endl;
                cin >> c;

                x[r][c] = '&';

        }

        for (int r = 0; r < 8; r++)
        {
          for (int c = 0; c < 8; c++)
          {
            if (x[r][c] == '&')
            {
              Check(x, r, c, check);
                cout << "check:" << check << endl;
                if (check == 1)
                {
                  cout<< "Error in cell of row: "<<r<< " and column: "<<c << endl;

                }


           }
         }
        }

        if (check == 0)
        {
                cout << "Good solution" << endl;
        }
}
```