CPTTRN1 - Character Patterns (Act 1)

[#basics](http://www.spoj.com/problems/tag/basics)

Using two characters: . (dot) and \* (asterisk) print a chessboard-like pattern. The first character printed should be \* (asterisk).

Input

You are given *t < 100* - the number of test cases and for each of the test cases two positive integers: *l* - the number of lines and *c* - the number of columns in the pattern (*l*, *c* < 100).

Output

For each of the test cases output the requested pattern (please have a look at the example). Use one line break in between successive patterns.

Example

**Input:**

3

3 1

4 4

2 5

**Output:**

\*

.

\*

\*.\*.

.\*.\*

\*.\*.

.\*.\*

\*.\*.\*

.\*.\*.

<http://www.spoj.com/problems/CPTTRN1/>

#include<stdio.h>

#include<stdlib.h>

#include <iostream>

#include <conio.h>

using namespace std;

void ajedrez(int f, int c)

{

    char m[f][c];

    bool flag = true;

    for (int i = 0; i < c; i++)

    {

        if (flag)

        {

            m[0][i] = '\*';

        }

        else

        {

            m[0][i] = '.';

        }

        flag = !flag;

    }

    flag = true;

    for (int i = 0; i < f; i++)

    {

        if (flag)

        {

            m[i][0] = '\*';

        }

        else

        {

            m[i][0] = '.';

        }

        flag = !flag;

    }

    for (int i = 1; i < f; i++)

    {

        for (int j = 1; j < c; j++)

        {

            if (m[i][j - 1] == '\*')

            {

                m[i][j] = '.';

            }

            else

            {

                m[i][j] = '\*';

            }

        }

    }

    for (int i = 0; i < f; i++)

    {

        for (int j = 0; j < c; j++)

        {

            //Console.Write(m[i, j]);

            printf("%c", m[i][j]);

        }

        //Console.WriteLine();

        printf("**\n**");

    }

}

int main(){

        int t, f, c;

        scanf("%d",&t);

        while(t--){

       scanf("%d %d",&f,&c);

       ajedrez(f,c);

    }

        getch();

        return 0;

}