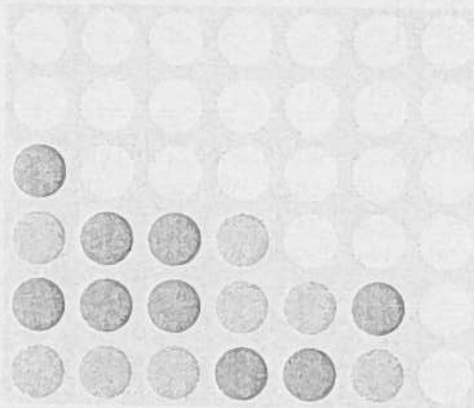


Write a test program that prompts the user to enter the number of rows and columns of a two-dimensional array then the values in the array, and displays true if the array contains four consecutive numbers with the same value. Otherwise, the program displays false. Here are some examples of the true cases:

0 1 0 3 1 6 1	0 1 0 3 1 6 1	0 1 0 3 1 6 1	0 1 0 3 1 6 1
0 1 6 8 6 0 1	0 1 6 8 6 0 1	0 1 6 8 6 0 1	0 1 6 8 6 0 1
5 6 2 1 8 2 9	5 5 2 1 8 2 9	5 6 2 1 6 2 9	9 6 2 1 8 2 9
6 5 6 1 1 9 1	6 5 6 1 1 9 1	6 5 6 6 1 9 1	6 9 6 1 1 9 1
1 3 6 1 4 0 7	1 5 6 1 4 0 7	1 3 6 1 4 0 7	1 3 9 1 4 0 7
3 3 3 3 4 0 7	3 5 3 3 4 0 7	3 6 3 3 4 0 7	3 3 3 9 4 0 7

*****8.20** (*Game: connect four*) Connect four is a two-player board game in which the players alternately drop colored disks into a seven-column, six-row vertically suspended grid, as shown below.



The objective of the game is to connect four same-colored disks in a row, a column, or a diagonal before your opponent can do likewise. The program prompts two players to drop a red or yellow disk alternately. In the preceding figure, the red disk is shown in a dark color and the yellow in a light color. Whenever a disk is dropped, the program redisplay the board on the console and determines the status of the game (win, draw, or continue). Here is a sample run:

```
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
```

Drop a red disk at column (0-6): 0

```
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| R | | | | |
| | | | | | |
```





Drop a yellow disk at column (0-6): 3

```

| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
|R| | |Y| | |

```

...

Drop a yellow disk at column (0-6): 6

```

| | | | | | |
| | | | | | |
| | | |R| | |
| | |Y|R|Y| |
| |R|Y|Y|Y|Y|
|R|Y|R|Y|R|R|

```

The yellow player won

*8.21

(*Central city*) Given a set of cities, the central city is the city that has the shortest total distance to all other cities. Write a program that prompts the user to enter the number of cities and the locations of the cities (coordinates), and finds the central city and its total distance to all other cities.



Enter the number of cities: 5

Enter the coordinates of the cities:

2.5 5 5.1 3 1 9 5.4 54 5.5 2.1

The central city is at (2.5, 5.0)

The total distance to all other cities is 60.81

*8.22

(*Even number of 1s*) Write a program that generates a 6-by-6 two-dimensional matrix filled with 0s and 1s, displays the matrix, and checks if every row and every column have an even number of 1s.

*8.23

(*Game: find the flipped cell*) Suppose you are given a 6-by-6 matrix filled with 0s and 1s. All rows and all columns have an even number of 1s. Let the user flip one cell (i.e., flip from 1 to 0 or from 0 to 1) and write a program to find which cell was flipped. Your program should prompt the user to enter a 6-by-6 array with 0s and 1s and find the first row r and first column c where the even number of the 1s property is violated (i.e., the number of 1s is not even). The flipped cell is at (r, c) . Here is a sample run:



VideoNote

Even number of 1s



Enter a 6-by-6 matrix row by row:

```

1 1 1 0 1 1 
1 1 1 1 0 0 
0 1 0 1 1 1 
1 1 1 1 1 1 
0 1 1 1 1 0 
1 0 0 0 0 1 

```

The flipped cell is at (0, 1)

*8.24

*8.25

*8.26

*8.27