Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
8199959	02:27:06	Magic Checkerboard	✔ **Accepted**	0.64 s	C++
	✔✔ ✔✔✔✔✔				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
checkerboard.cpp	4388 bytes	6c19d2b9fc83f6c7a7efe647c12af2b210c17c4e	download

Edit and resubmit this submission.

checkerboard.cpp

```
#include <bits/stdc++.h>

using namespace std;

long long f1(vector<vector<long long>>& board);
long long f2(vector<vector<long long>>& board);
void assignTop(vector<vector<long long>>& board, long long r, long long c, long long& val);
void assignLeft(vector<vector<long long>>& board, long long r, long long c, long long& val);

    Help   g findMinSum(vector<vector<long long>> board, int opc){
           , long s = 0,t=0,l=0;
    for(int i=0;i<board.size();i++){
```

```
        for(int j=0;j<board[0].size();j++){
            if(opc == 1){ //findMinSumIfEvenEven
                assignTop(board, i, j, t);
                assignLeft(board, i, j, l);

                if(board[i][j] == 0){
                    board[i][j] = max(t, l) + 1;
                    if(board[i][j] % 2 != i%2){
                        board[i][j]++;
                    }
                }
                else if( i%2!=board[i][j]%2 || board[i][j] <= l || board[i][j] <= t){
                    return LLONG_MAX;
                }
            }
            else if(opc == 2){ //findMinSumIfOddEven
                assignTop(board, i, j, t);
                assignLeft(board, i, j, l);
                if(board[i][j] == 0){
                    board[i][j] = max(t, l) + 1;
                    if(board[i][j]%2 == j%2){
                        board[i][j]++;
                    }
                }
                else if( j%2==board[i][j]%2 || board[i][j] <= l || board[i][j] <= t){
                        return LLONG_MAX;
                }
            }
            else if(opc == 3){ //findMinSumIfEvenOdd
                assignTop(board, i, j, t);
                assignLeft(board, i, j, l);
                if(board[i][j] == 0){
                    board[i][j] = max(t, l) + 1;
                    if(board[i][j] % 2 != j%2){
                        board[i][j]++;
                    }
                }
                else if( j%2!=board[i][j]%2 || board[i][j] <= l || board[i][j] <= t){
                        return LLONG_MAX;
                }
            }
            else{ //findMinSumIfOddOdd
                assignTop(board, i, j, t);
                assignLeft(board, i, j, l);
```

Help

```
                if(board[i][j] == 0){
                    board[i][j] = max(t, l) + 1;
                    if(board[i][j] % 2 == i%2){
                        board[i][j]++;
                    }
                }
                else if( i%2==board[i][j]%2 || board[i][j] <= l || board[i][j] <= t){
                        return LLONG_MAX;
                }
            }


            s += board[i][j];
        }
    }
    return s;
}



int main(){
    long long r = 0, c = 0;
    cin>>r>>c;
    vector<vector<long long>> tablero(r,vector<long long>(c));
    for(long long i=0;i<r;i++){
        for(long long j=0;j<c;j++){
            cin >> tablero[i][j];
        }
    }
    if(r==1)
        cout << f1(tablero);
    else if(c == 1)
        cout << f2(tablero);
    else{
        long long a = min(findMinSum(tablero,1), findMinSum(tablero,3));
        long long b = min(findMinSum(tablero,4), findMinSum(tablero,2));
        long long c = min(a, b);
        if(c == LLONG_MAX){
            cout<<-1;
        }
        else{
            cout<<c;
        }
    }
```

```
}


long long f1(vector<vector<long long>>& board){
    long long s = 0;
    for(long long i=0;i<board[0].size();i++){
        if(board[0][i] == 0){
            if(i == 0)
                board[0][i] = 1;
            else
                board[0][i] = board[0][i-1] + 1;
        }
        else{
            if(i!=0 && board[0][i] <= board[0][i-1])
                return -1;
        }
        s+=board[0][i];
    }
    return s;
}

long long f2(vector<vector<long long>>& board){
    long long s = 0;
    for(long long i=0;i<board.size();i++){
        if(board[i][0] == 0){
            if(i == 0)
                board[i][0] = 1;
            else
                board[i][0] = board[i-1][0] + 1;
        }
        else{
            if(i!=0 && board[i][0] <= board[i-1][0])
                return -1;
        }
        s+=board[i][0];
    }
    return s;
}

        ignTop(vector<vector<long long>>& board, long long r, long long c, long long& val){
Help     == 0)
        val = 0;
    else
```

```
145         val = board[r-1][c];
146 
147 }
148 
149 void assignLeft(vector<vector<long long>>& board, long long r, long long c, long long& val){
150     if(c == 0)
151         val = 0;
152     else
153         val = board[r][c-1];
154 }
```