

Submission

[illegible]

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

```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 long long f1(vector<vector<long long>>& board);
6 long long f2(vector<vector<long long>>& board);
7 void assignTop(vector<vector<long long>>& board, long long r, long long c, long long& val);
8 void assignLeft(vector<vector<long long>>& board, long long r, long long c, long long& val);
9
10     g findMinSum(vector<vector<long long>> board, int opc){
11         Help , long s = 0,t=0,l=0;
12         for(int i=0;i<board.size();i++){

```

```

13 for(int j=0;j<board[0].size();j++){
14     if(opc == 1){ //findMinSumIfEvenEven
15         assignTop(board, i, j, t);
16         assignLeft(board, i, j, l);
17
18         if(board[i][j] == 0){
19             board[i][j] = max(t, l) + 1;
20             if(board[i][j] % 2 != i%2){
21                 board[i][j]++;
22             }
23         }
24         else if( i%2!=board[i][j]%2 || board[i][j] <= 1 || board[i][j] <= t){
25             return LLONG_MAX;
26         }
27     }
28     else if(opc == 2){ //findMinSumIfOddEven
29         assignTop(board, i, j, t);
30         assignLeft(board, i, j, l);
31         if(board[i][j] == 0){
32             board[i][j] = max(t, l) + 1;
33             if(board[i][j]%2 == j%2){
34                 board[i][j]++;
35             }
36         }
37         else if( j%2==board[i][j]%2 || board[i][j] <= 1 || board[i][j] <= t){
38             return LLONG_MAX;
39         }
40     }
41     else if(opc == 3){ //findMinSumIfEvenOdd
42         assignTop(board, i, j, t);
43         assignLeft(board, i, j, l);
44         if(board[i][j] == 0){
45             board[i][j] = max(t, l) + 1;
46             if(board[i][j] % 2 != j%2){
47                 board[i][j]++;
48             }
49         }
50         else if( j%2!=board[i][j]%2 || board[i][j] <= 1 || board[i][j] <= t){
51             return LLONG_MAX;
52         }
53     }
54     else{ //findMinSumIfOddOdd
55         assignTop(board, i, j, t);
56         assignLeft(board, i, j, l);

```

```

57         if(board[i][j] == 0){
58             board[i][j] = max(t, 1) + 1;
59             if(board[i][j] % 2 == i%2){
60                 board[i][j]++;
61             }
62         }
63         else if( i%2==board[i][j]%2 || board[i][j] <= 1 || board[i][j] <= t){
64             return LLONG_MAX;
65         }
66     }
67
68     s += board[i][j];
69 }
70 }
71 }
72 return s;
73 }
74
75
76
77 int main(){
78     long long r = 0, c = 0;
79     cin>>r>>c;
80     vector<vector<long long>> tablero(r,vector<long long>(c));
81     for(long long i=0;i<r;i++){
82         for(long long j=0;j<c;j++){
83             cin >> tablero[i][j];
84         }
85     }
86     if(r==1)
87         cout << f1(tablero);
88     else if(c == 1)
89         cout << f2(tablero);
90     else{
91         long long a = min(findMinSum(tablero,1), findMinSum(tablero,3));
92         long long b = min(findMinSum(tablero,4), findMinSum(tablero,2));
93         long long c = min(a, b);
94         if(c == LLONG_MAX){
95             cout<<-1;
96         }
97         else{
98             cout<<c;
99         }
100     }

```

```

101 }
102
103
104
105 long long f1(vector<vector<long long>>& board){
106     long long s = 0;
107     for(long long i=0;i<board[0].size();i++){
108         if(board[0][i] == 0){
109             if(i == 0)
110                 board[0][i] = 1;
111             else
112                 board[0][i] = board[0][i-1] + 1;
113         }
114         else{
115             if(i!=0 && board[0][i] <= board[0][i-1])
116                 return -1;
117         }
118         s+=board[0][i];
119     }
120     return s;
121 }
122
123 long long f2(vector<vector<long long>>& board){
124     long long s = 0;
125     for(long long i=0;i<board.size();i++){
126         if(board[i][0] == 0){
127             if(i == 0)
128                 board[i][0] = 1;
129             else
130                 board[i][0] = board[i-1][0] + 1;
131         }
132         else{
133             if(i!=0 && board[i][0] <= board[i-1][0])
134                 return -1;
135         }
136         s+=board[i][0];
137     }
138     return s;
139 }
140
141     ignTop(vector<vector<long long>>& board, long long r, long long c, long long& val){
142 Help     == 0)
143         val = 0;
144     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 }
```