

Clean Robot

We have to plan a path for a cleaning robot to clean a rectangular room floor of size $N \times M$. The room floor paved with square tiles whose size fits the cleaning robot (1×1). There are **clean tiles** and **dirty tiles**, and the robot can change a dirty tile to a clean tile by visiting the tile. Also there may be some **obstacles** (furniture) whose size fits a tile in the room. If there is an obstacle on a tile, the robot **cannot visit** it. The robot moves to an adjacent tile with one move. The tile onto which the robot moves must be one of four tiles (i.e., east, west, north or south) adjacent to the tile where the robot is present. The robot may visit a tile **twice** or more.

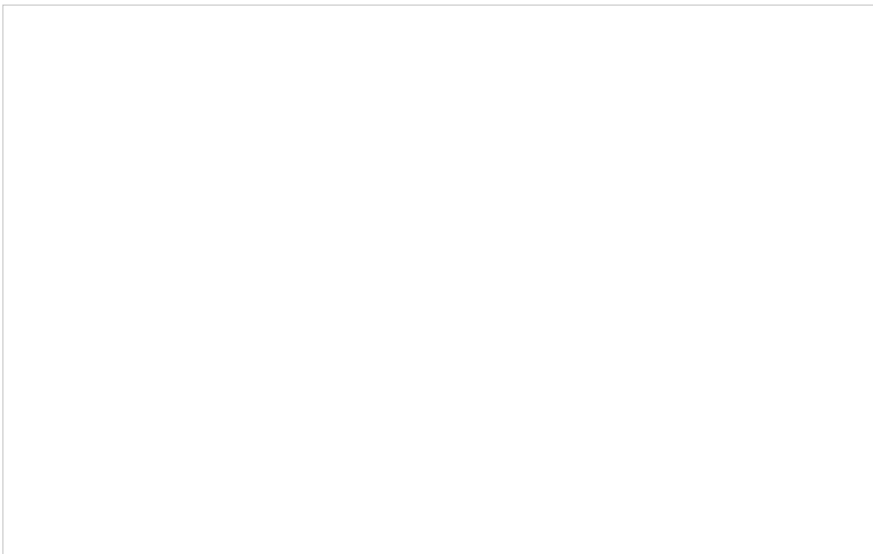
Your task is to write a program which computes the minimum number of moves for the robot to change all dirty tiles to clean tiles, if ever possible.

Time limit: 1s (C/C++), 2s (Java)

Submit limit: 10 times

Example:

The following is a room of size 5×7 , with 3 dirty tiles, and 0 furniture. The answer for this case is 8.



Input

The input consists of multiple maps, the first line is the number of test case T ($T \leq 50$).

Each test case begins with N and M representing the size of the room. ($5 \leq N$, $M \leq 100$)

The next N line representing the arrangement of the room with following describe:

- 0 : a clean tile
- 1 : a dirty tile
- 2 : a piece of furniture (obstacle)
- 3 : the robot (initial position)

Trong bản đồ số lượng ô bản **không quá 10** và **chỉ có một người máy** .

Đầu ra

In mỗi trường hợp thử nghiệm trên hai dòng, dòng đầu tiên của mỗi trường hợp thử nghiệm là "Case #x", trong đó x là số trường hợp thử nghiệm. Dòng tiếp theo là số lần di chuyển tối thiểu để robot đổi hết gạch bẩn thành gạch sạch. Nếu bản đồ bao gồm các ô bản mà rô bốt không thể tiếp cận, chương trình của bạn sẽ xuất ra -1.

Vật mẫu

Đầu vào

```
5
5 7
0 0 0 0 0 0 0
0 3 0 0 0 1 0
0 0 0 0 0 0 0
0 1 0 0 0 1 0
0 0 0 0 0 0 0
5 15
0 0 0 0 2 0 2 0 0 0 0 1 2 0 1
0 0 0 1 0 2 0 2 2 0 1 2 0 0 0
2 1 0 2 0 1 0 2 0 0 0 0 0 0 0
0 0 0 1 0 2 0 0 1 2 0 0 2 0 0
0 2 1 0 2 0 0 0 0 0 3 0 0 0 0
.....
```

Đầu ra

```
Trường hợp 1
số 8
Trường hợp số 2
38
Trường hợp số 3
37
Trường hợp số 4
-1
Trường hợp số 5
49
```