

## G: Tron Garbage Collector

Source file name: `garbage.c`, `garbage.cpp`, `garbage.java`, or `garbage.py`

Author: Rafael García

According to Wikipedia, a *garbage collector* in computer science is

... a form of automatic memory management. It attempts to reclaim garbage or memory occupied by objects that are no longer in use by the program.

How does a memory garbage collector work? Think of the main memory as the Squareland: a large grid with dimensions  $R$  and  $C$  where the lines represent streets and the boxes are labeled with the number of data to be collected. A car is driven through the streets to remove the data adjacent to each street segment it visits. More precisely, a number in a box identifies the number of times such a box must be visited by the car in order to collect the garbage (i.e., the number of times an adjacent street segment needs to be visited by the car). If such a number is 0, then the box can not be visited. The labels 1, 2, 3, and 4 indicate that a box must be visited 1, 2, 3, or 4 times, respectively. Otherwise, the box can be visited any number of times. However, there are additional restrictions: the car route must start and end at the same street intersection and it cannot visit a corner (except for the first one) more than once.

For example, consider the following  $5 \times 5$  grid representing Squareland:

2		2		3
				3
2			1	
	3	2	0	
		1	2	2

The following is a possible route for the garbage collecting car:

2		2		3
				3
2			1	
	3	2	0	
		1	2	2

In order to design the garbage collection task, you are required to write a program to determine if there exists a route that can collect all the garbage satisfying the constraints above-mentioned.

## Input

The input consists of several test cases. The first line in a test case contains two blank-separated integers  $0 < R, C < 6$ . Each of the next  $R$  lines contains  $C$  symbols denoting the number of data values that must be eliminated from the grid:

- If the number of data values that must be eliminated is  $d$ , then the number  $d$  ( $0 \leq d \leq 4$ ) appears.
- If the collector can visit the streets adjacent to a box but it is not required to remove anything, then the dot (.) appears.

The input ends with two blank-separated zeroes.

*The input must be read from standard input.*

## Output

For each test case, output YES if it is possible to find a route under the given constraints and NO otherwise.

*The output must be written to standard output.*

Sample Input	Sample Output
5 5 2.2.3 . . . . 3 2. . 1. . 320. . . 122 5 5 2.2.3 . 4. . 3 2. . 1. . 320. . . 122 0 0	YES NO