



## Flappy Bird

Task ID: flappy	Time limit: 1.0 sec	Memory limit: 64 MB
-----------------	---------------------	---------------------

Bist du schon süchtig nach dem Spiel *Flappy Bird*? Du kannst das Spiel selber im Netz spielen unter <http://flappybird.io/>. Doch vergiss nicht zuerst die Aufgabe zu lösen!

Das Ziel des Spieles ist es einen kleinen Vogel über ein Spielfeld von links nach rechts zu steuern. Es gibt nur eine Entscheidung, die du in jedem Schritt fällen musst: flattern oder nicht flattern. Durch flattern oder nicht flattern, kann der Vogel höher oder tiefer fliegen währenddem er immerzu nach rechts fliegt. Das Spielfeld wird durch eine binäres Gitter mit  $H$  Zeilen und  $W$  Spalten beschrieben. 1 stellt ein Hindernis dar und 0 steht für freien Luftraum.

In jedem Zeitschritt bewegt sich der Vogel eine Spalte nach rechts. Die Anzahl Zeilen, welche er sich nach oben oder unten bewegt hängt davon ab, wie viel Zeit seit dem letzten Flügelschlag vergangen ist. Wenn er während dem Zeitschritt flattert, bewegt er sich zwei Zeilen nach oben. Falls er im letzten Schritt flatterte, steigt er um eine Zeile und so weiter. Die Tabelle unten beschreibt das Verhalten des Vogels:

Zeit seit dem letzten Flügelschlag	0	1	2	3	4	$\geq 5$
Vertikale Verschiebung in diesem Schritt	+2	+1	0	-1	-2	-4

Der Vogel kann seine Flügel in jedem beliebigen Zeitschritt flattern lassen und er kann auf einer beliebigen Höhe auf der linken Seite starten. Deine Aufgabe ist es zu entscheiden, ob es möglich ist in den richtigen Momenten zu flattern, sodass der Vogel bis ganz nach rechts fliegen kann ohne je in ein Hindernis zu fliegen. Am Anfang nehmen wir an, dass es sehr lange her ist seit dem letzten Flügelschlag, sodass der Vogel vier Zeilen pro Zeitschritt fällt, bis er zum ersten Mal die Flügel schlägt. Der Vogel muss zu jeder Zeit auf der Spielkarte bleiben. Er soll also nie unter den Boden oder über die Decke hinaus fliegen.

Du kannst annehmen, dass der Vogel sich in jedem Zeitschritt instantan bewegt. Der Vogel muss sich also nur zu ganzzahligen Zeitwerten auf freien Positionen befinden und es spielt keine Rolle ob es Hindernisse auf dem Weg dahin gibt, vgl. dazu die Beispiele unten.

### Eingabe

Die erste Zeile der Eingabe besteht aus zwei Ganzzahlen  $W$  und  $H$ , der Anzahl Spalten und Zeilen des Spielfeldes. Die folgenden  $H$  Zeilen enthalten je  $W$  Ziffern, alle entweder 0 oder 1 um das Spielfeld zu beschreiben, 0 repräsentiert freien Luftraum und 1 steht für Hindernisse. Du kannst annehmen, dass die erste und letzte Zeile nur aus 1en besteht, was den Boden und die Decke darstellt. Ausserdem kannst du annehmen, dass es mindestens eine 0 in der ersten Spalte gibt, einen möglichen Ort für den Vogel um seine Reise zu starten.

### Ausgabe

Du sollst eine Zeile mit einem einzelnen Wort ausgeben: YES falls es möglich ist, dass der Vogel auf die andere Seite fliegt und NO sonst.

LIMITS:

Es gibt fünf Testgruppen, jede gibt 20 Punkte.



- In der ersten Gruppe gilt  $3 \leq H \leq 8$ ,  $5 \leq W \leq 20$  und alle Hindernisse sind nur in der ersten und letzten Zeile (so wie im letzten Beispiel).
- In der zweiten und dritten Gruppe gilt  $5 \leq H \leq 50$ ,  $10 \leq W \leq 100$ .
- In der vierten und fünften Gruppe gilt  $50 \leq H \leq 500$ ,  $50 \leq W \leq 2000$ .

input	output
<pre>11 8 11111111111 11111110011 01111101101 11100111111 11011011110 11111111111 10111111111 11111111111</pre>	YES

*Erklärung:* In diesem Beispiel ist alles ausser die Positionen, durch die der Vogel fliegt mit Hindernissen belegt. Um die andere Seite zu erreichen, schlägt der Vogel seine Flügel genau zwei Mal. Zuerst wenn er von der zweiten in die dritte Spalte fliegt und später wenn er von der sechsten in die siebte Spalte fliegt. Das heisst er fällt zuerst vier Zeilen, schlägt dann mit den Flügeln und steigt zuerst zwei, dann eine und keine Zeile. Darauf fällt er um eine Zeile, schlägt mit den Flügeln um wieder zwei anzusteigen und so weiter.

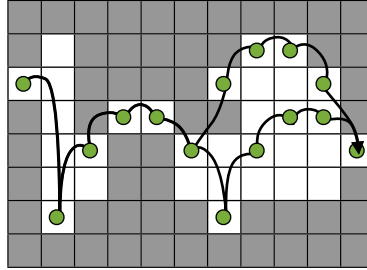
Die Tabelle unten zeigt den Pfad zusammen mit der Anzahl Schritte seit dem letzten Flügelschlag:

```
.....
.....12..
5.....0..3.
...12.....
..0..3....4
.....
.5.....
.....
```

input	output
<pre>11 8 11111111111 10111110011 00111100001 10100110001 10011000000 10011000001 10111101111 11111111111</pre>	YES



*Erklärung:* In diesem zweiten Spielfeld hat es ein wenig mehr freien Platz als im ersten Beispiel, weshalb es hier zwei Möglichkeiten gibt, um auf die andere Seite zu gelangen. Beide



Wege sind in der Abbildung unten gezeigt:

input

output

```

6 5
111111
000000
000000
000000
111111
    
```

```

NO
    
```

*Erklärung:* Egal wo der Vogel startet, muss er im ersten Schritt die Flügel schlagen um zu verhindern, dass er auf den Boden fällt. Aber dann wird er im zweiten oder dritten Schritt die Decke erreichen, weshalb es keine Möglichkeit gibt um sicher auf die andere Seite zu gelangen.