



Practice Arena

Practice problems aimed to improve your coding skills.

- 📁 PRACTICE-02_SCAN-PRINT
- 📁 PRACTICE-03_TYPES
- 📁 LAB-PRAC-02_SCAN-PRINT
- 📁 LAB-PRAC-01
- 📁 PRACTICE-04_COND
- 📁 BONUS-PRAC-02
- 📁 LAB-PRAC-03_TYPES
- 📁 PRACTICE-05_COND-LOOPS
- 📁 LAB-PRAC-04_COND
- 📁 LAB-PRAC-05_CONDDOOPS
- 📁 PRACTICE-07_LOOPS-ARR
- 📁 LAB-PRAC-06_LOOPS
- 📁 LAB-PRAC-07_LOOPS-ARR
 - ❓ Home Alone
 - ❓ Arrangements with Arrays
 - ❓ Overlapping Patterns
 - ❓ Lucky Draw
 - ❓ Diamond Array
 - ❓ Linear Leap
 - ❓ Candy Crush
 - ❓ Nested Safes
 - ❓ Heros Arc
 - ❓ Linear Loopy Maze
 - ❓ Histogram Heights
 - ❓ Changing Times
- 📁 LABEXAM-PRAC-01_MIDSEM
- 📁 PRACTICE-09_PTR-MAT
- 📁 LAB-PRAC-08_ARR-STR
- 📁 PRACTICE-10_MAT-FUN
- 📁 LAB-PRAC-09_PTR-MAT
- 📁 LAB-PRAC-10_MAT-FUN
- 📁 PRACTICE-11_FUN-PTR
- 📁 LAB-PRAC-11_FUN-PTR
- 📁 LAB-PRAC-12_FUN-STRUC
- 📁 LABEXAM-PRAC-02_ENDSEM
- 📁 LAB-PRAC-13_STRUC-NUM
- 📁 LAB-PRAC-14_SORT-MISC

Linear Leap

LAB-PRAC-07_LOOPS-ARR

Linear Leap [20 marks]

Problem Statement

You will be given as input, a list of **non-negative integers**. The list will end with a -1 but the -1 is not a part of the list. The value of a number in this list denotes its "jump size". For example, if the number at location i is 3, then the jump size at that element is 3. This means that all locations to the right of that location, within a distance of 3, are reachable from that location in a single jump. More precisely, in a single "jump" Mr C can go from location i to location $i+1$ or $i+2$ or $i+3$ if the number at location i is 3. The first number of the list is said to be at location 1 (not like arrays where the first element is subscript 0).

The number at location j will be said to be "reachable" from location i if it can be reached in any number of jump. Mr C starts by jumping on location 1 (this is his initial jump). He can now take any number of jumps, in any order, as allowed by the above rules. However, **all jumps are made to the right of the list i.e. no backward jumps**.

Print, as your output "YES" (without quotes) if the last location in the list (i.e. the location just before -1 is there) is reachable from location 1, else print "NO" (without quotes).

Caution

1. Be careful about extra/missing lines and extra/missing spaces in your output.
 2. **Any hardcoding attempts will be given a straight 0**
-

EXAMPLE 1:

INPUT

3 1 2 0 0 -1

OUTPUT:

YES

Explanation: Since the number at location 1 is 3, the 2nd, 3rd and 4th locations can be reached in a single step from location 1. Now the last location (i.e. location number 5) can be reached in a single step from location 3 since the number stored at location 3 is 2. Thus, the last location before -1 is reachable from location 1.

Step 1: Land on location 1

Step 2: Jump to location 3

Step 3: Jump to location 5

Note that Mr C could have jumped to location 4 from location 1 as well but that would have been useless since he would have been stuck there since the number stored there is 0. However, the answer is YES since there does exist an alternate set of jumps that do take Mr C to the last location before the -1.

EXAMPLE 2:

INPUT

3 1 1 0 1 2 -1

OUTPUT:
NO

Explanation: Since the number at location 1 is 3, the locations 2, 3, 4 can be reached in a single jump from location 1. Now lets see what happens on jumping to each of these locations

1. The number at location 2 is 1 so only location 3 can be reached in a single jump from location 2
2. The number at location 3 is 2 so only location 4 can be reached in a single jump from location 3
3. The number at location 4 is zero so if we land there, we are stuck.

Thus, we see that at most, we can reach location 4 from location 1 in many ways (1 => 4, 1 => 3 => 4, 1 => 2 => 3 => 4) but we always get stuck there. Thus, there is no way we can reach the location 6 which is the last location before -1. So the answer is NO.

Grading Scheme:

Total marks: **[20 Points]**

There will be no partial grading in this question. An exact match will receive full marks whereas an incomplete match will receive 0 points. Please be careful of missing/extra spaces and missing/lines (take help of visible test cases). Each visible test case is worth 1 point and each hidden test case is worth 2 points. There are 2 visible and 4 hidden test cases.

 **Start Solving!** (/editor/practice/6142)