**Jump Out**

Max. Marks: 100

You have a field of N boosters. The power of ith booster is given as Ai. If you stand on ithbooster, you will get the power Ai and you can make a jump of length Ai. That is, if you are standing on 3rd booster and it's power is 5, then you will land on position 8. Currently, you are standing outside the field to the left of first booster and you want to across this field by reaching to the right of Nth booster. You want to make only one jump to any booster such that you will finally land to right of Nth booster. Print the minimum length of initial jump you should make such that you will finally land outside the field by using exactly one booster.

**INPUT:**  
First line of input consists of integer **N**. Next line consists of N integers A1, A2...AN.

**OUTPUT:**   
Output the minimum length of jump you should make such that you will finally land outside the field using exactly 1 booster.

**CONSTRAINTS:**   
1 ≤ N ≤ 104  
1 ≤ Ai ≤ 104

**SAMPLE INPUT**

5

4 2 4 2 3

**SAMPLE OUTPUT**

3

**Explanation**

You can cross the field by making the first jump of length 3, 4, or 5. 3 is the answer since we want smallest initial jump.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded if any testcase passes.

**Allowed Languages:**C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Visual Basic, Kotlin

<https://www.hackerearth.com/challenge/competitive/june-circuits-17/algorithm/jump-out-34/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

int[] arr = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

//int[] arr = { 4, 2, 4, 2, 3 };

for (int i = 0; i < arr.Length; i++)

{

if (i + 1 + arr[i] > n)

{

Console.WriteLine(i+1);

break;

}

}

Console.ReadLine();

}

}

}