

Meera teaches a class of n students, and every day in her classroom is an adventure. Today is drawing day!

The students are sitting around a round table, and they are numbered from 1 to n in the *clockwise* direction. This means that the students are numbered $1, 2, 3, \dots, n-1, n$, and students 1 and n are sitting next to each other.

After letting the students draw for a certain period of time, Meera starts collecting their work to ensure she has time to review all the drawings before the end of the day. However, some of her students aren't finished drawing! Each student i needs t_i extra minutes to complete their drawing.

Meera collects the drawings sequentially in the clockwise direction, starting with student ID x , and it takes her exactly 1 minute to review each drawing. This means that student x gets 0 extra minutes to complete their drawing, student $x+1$ gets 1 extra minute, student $x+2$ gets 2 extra minutes, and so on. *Note* that Meera will still spend 1 minute for each student even if the drawing isn't ready.

Given the values of t_1, t_2, \dots, t_n , help Meera choose the best possible x to start collecting drawings from, such that the number of students able to complete their drawings is maximal. Then print x on a new line. If there are multiple such IDs, select the smallest one.

Input Format

The first line contains a single positive integer, n , denoting the number of students in the class. The second line contains n space-separated integers describing the respective amounts of time that each student needs to finish their drawings (i.e., t_1, t_2, \dots, t_n).

Constraints

- $1 \leq n \leq 10^5$
- $0 \leq t_i \leq n$

Subtasks

- $1 \leq n \leq 10^4$ for **30%** of the maximum score.

Output Format

Print an integer denoting the ID number, x , where Meera should start collecting the drawings such that a maximal number of students can complete their drawings. If there are multiple such IDs, select the smallest one.