2 The Sleepy Employee

You have just moved to Washington, DC for your new job. The city is so crowded that if you leave for work less than m minutes before your company opens, you will get stuck in heavy traffic and will arrive late. You do not know anything about the value of m, except that it is an integer between 1 and a given number n, and it is the same every day.

You use a popular ride service to get to work, and their surge pricing policy mean that the cost of going to work depends on the time that you leave home. You have access to a price list, $P[1..n]$, where $P[i]$ is the price that you pay to ride to work assuming that you leave i minutes before your company opens. If you arrive late to work, your company fines you \$10 dollars, and if you arrive on time or early, your company rewards you with a free cup of coffee, valued at \$2.

You decide to determine a strategy that will tell you the value of m. Each day, the strategy indicates a time i minutes before opening time, where $1 \leq i \leq n$. You pay $P[i]$ to ride to work. If you arrive on time, you earn \$2 and have learned that $m \leq i$. If you arrive late, you lose \$10, but have learned that $m > i$. After a sufficient number of trials, you can infer the exact value of m. Your objective is to generate a strategy that determines the value of m at the lowest cost. Since the value of m is unknown, the cost of a strategy is the maximum cost that you pay, under the assumption that m can take on any value between 1 and n. Your objective is to compute the strategy that has the minimum worst-case cost, based on the price list $P[1..n]$.

All quantities are positive integers. You may assume that $n \leq 10^4$, and $P[i] > \$2$, for all i.

Input and output files can be found at: `http://challengebox.cs.umd.edu/2019/Sleepy`

Input:

The first line contains the value n, and the next n lines contain the values $P[i]$, for $i = 1, 2, \ldots, n$.

Output:

A single integer indicating the minimum worst-case cost of the optimum strategy.

Example:

Suppose that $n = 3$ and $P[1..3] = \{30, 5, 20\}$. There is no benefit to testing $i = 3$, because you know that you will arrive on time. Between the options of testing $i = 1$ or $i = 2$ first, the latter is preferred. This is because if you test $i = 1$ first, you pay the \$30 for the ride and if you arrive late, you will suffer the \$10 fee and will still need to test $i = 2$. On the other hand, if you test $i = 2$ first, if you are late you do not need to test $i = 1$. If you are on time, you will need to test $i = 1$, but you obtain the \$2 reward. The worst case for this strategy arises when $m = 2$. In this case you test $i = 2$, arrive on time and then test $i = 1$ and arrive late. The total cost is $\$5 - \$2 + \$30 + \$10 = \$43$. (This is the worst case, as all other values of m lead to lower costs.)

Input:	Output:
3	43
30	
5	
20	