

## B2. Military Trainings

time limit per test: 3 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

The Smart Beaver from ABBYY started cooperating with the Ministry of Defence. Now they train soldiers to move armoured columns. The training involves testing a new type of tanks that can transmit information. To test the new type of tanks, the training has a special exercise, its essence is as follows.

Initially, the column consists of  $n$  tanks sequentially numbered from 1 to  $n$  in the order of position in the column from its beginning to its end. During the whole exercise, exactly  $n$  messages must be transferred from the beginning of the column to its end.

Transferring one message is as follows. The tank that goes first in the column transmits the message to some tank in the column. The tank which received the message sends it further down the column. The process is continued until the last tank receives the message. It is possible that not all tanks in the column will receive the message — it is important that the last tank in the column should receive the message.

After the last tank (tank number  $n$ ) receives the message, it moves to the beginning of the column and sends another message to the end of the column in the same manner. When the message reaches the last tank (tank number  $n - 1$ ), that tank moves to the beginning of the column and sends the next message to the end of the column, and so on. Thus, the exercise is completed when the tanks in the column return to their original order, that is, immediately after tank number 1 moves to the beginning of the column.

If the tanks were initially placed in the column in the order  $1, 2, \dots, n$ , then after the first message their order changes to  $n, 1, \dots, n - 1$ , after the second message it changes to  $n - 1, n, 1, \dots, n - 2$ , and so on.

The tanks are constructed in a very peculiar way. The tank with number  $i$  is characterized by one integer  $a_i$ , which is called the message receiving radius of this tank.

Transferring a message between two tanks takes one second, however, not always one tank can transmit a message to another one. Let's consider two tanks in the column such that the first of them is the  $i$ -th in the column counting from the beginning, and the second one is the  $j$ -th in the column, and suppose the second tank has number  $x$ . Then the first tank can transmit a message to the second tank if  $i < j$  and  $i \geq j - a_x$ .

The Ministry of Defense (and soon the Smart Beaver) faced the question of how to organize the training efficiently. The exercise should be finished as quickly as possible. We'll neglect the time that the tanks spend on moving along the column, since improving the tanks' speed is not a priority for this training.

You are given the number of tanks, as well as the message receiving radii of all tanks. You must help the Smart Beaver and organize the transferring of messages in a way that makes the total transmission time of all messages as small as possible.

### Input

The first line contains integer  $n$  — the number of tanks in the column. Each of the next  $n$  lines contains one integer  $a_i$  ( $1 \leq a_i \leq 250000$ ,  $1 \leq i \leq n$ ) — the message receiving radii of the tanks in the order from tank 1 to tank  $n$  (let us remind you that initially the tanks are located in the column in ascending order of their numbers).

To get the full points for the first group of tests it is sufficient to solve the problem with  $2 \leq n \leq 300$ .

To get the full points for the second group of tests it is sufficient to solve the problem with  $2 \leq n \leq 10000$ .

To get the full points for the third group of tests it is sufficient to solve the problem with  $2 \leq n \leq 250000$ .

### Output

Print a single integer — the minimum possible total time of transmitting the messages.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
3 2 1 1
output
5

input
5 2 2 2 2 2 2
output
10

Note

In the first sample the original order of tanks is 1, 2, 3. The first tank sends a message to the second one, then the second tank sends it to the third one — it takes two seconds. The third tank moves to the beginning of the column and the order of tanks now is 3, 1, 2. The third tank sends a message to the first one, then the first one sends it to the second one — it takes two more seconds. The second tank moves to the beginning and the order of the tanks is now 2, 3, 1. With this arrangement, the second tank can immediately send a message to the first one, since the message receiving radius of the first tank is large enough — it takes one second. Finally, the tanks return to their original order 1, 2, 3. In total, the exercise takes 5 seconds.

In the second sample, all five tanks are the same and sending a single message takes two seconds, so in total the exercise takes 10 seconds.