
Bus number 74

P55376_en

Every evening, Professor Oak walks from his job to the bus stop, waits for a bus 74, and then rides it home. But he has lately noticed that, usually, several 74's have just passed when he reaches the bus stop, so he has to wait for a long time for the next 74. Prof. Oak, who is a bit paranoid, starts thinking that there is a conspiracy against him. Funny, isn't it?

Surprisingly, Prof. Oak is right. The same moment he leaves his job, a geostationary satellite contacts all the 74's. From that moment on, every 74 chooses its speed so that Prof. Oak has to wait for the maximum time at the bus stop. During this time, the 74's don't attend the rest of users, and do not stop no matter what, red signals included. There is just one exception to this rule: since buses have just one lane, they cannot overtake (nor overlap) each other.

Let's formalize more the problem you must solve. Let t be the time for Prof. Oak to reach the bus stop, let m be the minimum speed for a 74, let M be the maximum speed for a 74, let L be the length of the bus lane, and let n be the number of 74's. Model the bus lane with the circular interval $[0, L)$. The bus stop is located at $[0, 1)$. Every 74 has length 1 as well. At time 0, each 74 numbered $i = 1, \dots, n$ is at some known interval $[p_i, p_i + 1)$. From that moment on, every 74 moves to the right (circularly, if the end is reached). From the moment t on, the first instant that the bus stop is overlapped by one or more 74's, Prof. Oak rides any of them. Your task is to compute the maximum possible waiting time.

Note: Every moment, every bus can choose any speed (fractional or not) between m and M , both included. Look at the sample input and sample output below for some limiting cases.

Input

Input is all integer numbers, and consists of several cases with t, m, M, L and n in this order, followed by p_1, \dots, p_n . Assume $t \geq 0, 1 \leq m \leq M, 1 \leq n \leq L$, that all the p_i 's are correct and different, and that no given number is larger than 10000.

Output

For every case, print with four digits after the decimal point the maximum waiting time.

Sample input 1

```
0 2 3 9 1 1
0 4 4 9 1 8
3 1 4 12 2 10 4
100 1 1 3 1 0
```

Sample output 1

```
3.5000
0.0000
9.0000
1.0000
```

Problem information

Author: Salvador Roura

Generation: 2026-01-25T11:19:28.432Z

© Jutge.org, 2006–2026.

<https://jutge.org>