

Traversal

I arrive at a big lake and I have to cross it. Luckily, I'm a very good jumper, but the lake is too big to be crossed in one jump. On the shore, I find **N** boxes of different heights, set in a certain order. If I throw a box into the lake, it will float and it will have the same height as on the shore. This is good, because I intend to throw some boxes into the lake and get from one shore to the other by jumping from box to box.

The only things to consider are:

- The lake is big, so I must throw at least 2 boxes, which means that in order to cross the lake I have to make at least 3 jumps.
- Not all the boxes have to be thrown; some of them may be ignored.
- The boxes can be thrown into the lake only in the order they are found on the shore and I have to jump on them in this order.
- The height difference between two consecutive boxes I use must be at most **H** meters, because I can jump a lot in length, but I have some problems with jumping in height.
- The height of a box doesn't change when I jump on it.
- I'm always able to jump from the shore to a box and from a box to the shore, no matter what the height of the box is.

Facing so many possibilities that respect the above conditions, I begin counting the number of possibilities that I have, instead of actually crossing the lake. I quickly find the answer and I wonder whether you can also find it as fast as I did.

Task

Write a program that determines the number of possibilities to cross the lake in the above conditions. Since the number can be quite big, you only have to output the remainder of this number, when divided by **9901**.

Input data

(file: **travers.in**)

The first line of the input file contains two integers **N** and **H**, separated by a space, representing the number of boxes and the maximum height difference between two consecutive boxes thrown into the lake. The following **N** lines contain the heights of the boxes, in the order the boxes are set on the shore. The **(i+1)th** line contains the height of the **ith** box.

Output data

(file: **travers.out**)

The output file consists of a single line, containing the number of possibilities modulo **9901**.

Constraints

- **1 < N < 100 001**
- **0 < H < 100 000 001**
- The height of any box is a strictly positive integer and does not exceed **100 000 000**
- 40% of the test cases will have the height of any box smaller than **10 001**

Example

travers.in	travers.out	<i>Explanation</i>
4 2	4	There are 4 possibilities:
1		1 3
3		1 3 5
7		3 5
5		7 5

Time limit: 0.5 seconds/test