Problem I. SPPPSPSS.

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

SPPPSPSS. stands for `Sort Permutation Performing Prefix Sort Plus Suffix Sort`.

You are given a permutation p of length n. You want to sort it in increasing order using the minimum number of operations. In the k-th operation you need to choose either the prefix of length k or the suffix of length k, and sort it in increasing order.

Input

The first line contains one integer n ($1 \leq n \leq 10^6$) — the size of the permutation.

The second line contains the permutation $p_1, p_2, \ldots, p_n$.

Output

Suppose the minimum number of operations needed to sort the given permutation is equal to m. Then you should print a string of length $m + 1$, the last character should be " .", and all other characters should be either "P" or "S" describing whether you want to sort prefix ("P") or suffix ("S") in the respective operation.

Examples

standard input	standard output
3 1 2 3	.
2 2 1	SP.
9 3 2 4 1 5 6 7 9 8	SSSP.
10 2 9 5 7 10 6 3 1 8 4	SPPPSPSS.

Note

This is how the permutation will change in the fourth sample:

Before	Operation	After
2 9 5 7 10 6 3 1 8 <u>4</u>	S : Sort suffix of length 1	2 9 5 7 10 6 3 1 8 <u>4</u>
<u>2 9</u> 5 7 10 6 3 1 8 4	P : Sort prefix of length 2	<u>2 9</u> 5 7 10 6 3 1 8 4
<u>2 9 5</u> 7 10 6 3 1 8 4	P : Sort prefix of length 3	<u>2 5 9</u> 7 10 6 3 1 8 4
<u>2 5 9 7</u> 10 6 3 1 8 4	P : Sort prefix of length 4	<u>2 5 7 9</u> 10 6 3 1 8 4
2 5 7 9 10 <u>6 3 1 8 4</u>	S : Sort suffix of length 5	2 5 7 9 10 <u>1 3 4 6 8</u>
<u>2 5 7 9 10 1</u> 3 4 6 8	P : Sort prefix of length 6	<u>1 2 5 7 9 10</u> 3 4 6 8
1 2 5 <u>7 9 10 3 4 6 8</u>	S : Sort suffix of length 7	1 2 5 <u>3 4 6 7 8 9 10</u>
1 2 5 <u>3 4 6 7 8 9 10</u>	S : Sort suffix of length 8	1 2 <u>3 4 5 6 7 8 9 10</u>