# D. Inna and Sequence

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dima's spent much time thinking what present to give to Inna and gave her an empty sequence $w$. Now they want to fill sequence $w$ with numbers zero and one. For that, they decided to play an amusing game.

Before the game begins, Dima chooses $m$ integers $a_1, a_2, ..., a_m$ $(1 \le a_1 < a_2 < ... < a_m)$. Then Inna and Dima start playing, that is, adding numbers to sequence $w$. Each new number they choose is added to the end of the sequence. At some moments of time Dima feels that the game is going to end too soon (and he wants to play with Inna as long as possible), so he hits a table hard with his fist. At that the $a_1$-th, $a_2$-th, $a_3$-th, ..., $a_k$-th numbers from the beginning simultaneously fall out of the sequence (the sequence gets $k$ numbers less). Here $k$ is such maximum number that value $a_k$ doesn't exceed the current length of the sequence. If number $a_1$ is larger than the current length of $w$, then nothing falls out of the sequence.

You are given the chronological sequence of events in the game. Each event is either adding a number to the end of sequence $w$ or Dima's hit on the table. Calculate the sequence $w$ after all these events happen.

## Input

The first line of the input contains two integers $n$ and $m$ $(1 \le n, m \le 10^6)$ showing how many events took place and how many numbers Dima chose.

The next line contains $m$ distinct integers $a_i$ $(1 \le a_i \le 10^6)$ sorted in the increasing order.

Next $n$ lines describe the events in the chronological order. Each line contains a single integer: -1, 0 or 1. Number -1 means that Dima hits the table. Number 0 means that Inna and Dima add number 0 to the end of the sequence. Number 1 means that Inna and Dima add number 1 to the end of the sequence.

## Output

In a single line print a sequence of numbers 0 and 1 — the elements of the sequence after all events happen. Print the elements of the sequence in the order from the beginning to the end of the sequence.

If after all events the sequence ends up **empty**, print "`Poor stack!`".

### Examples

| input |
| --- |
| 10 3<br>1 3 6<br>-1<br>1<br>1<br>0<br>0<br>-1<br>0<br>1<br>-1<br>1 |

| output |
| --- |
| 011 |

| input |
| --- |
| 2 1<br>1 |

```
1
-1
```

**output**

```
Poor stack!
```