Problem B
Berry Battle 2



Picture by MAKY.OREL, public domain

Erik and his grandpa often go to the forest to pick blueberries. Grandpa always finds the most berries, and even though it is not a competition, Erik has had enough of this. He has observed that grandpa uses a simple greedy strategy to pick as many berries as possible. With this information, Erik will try to finally pick at least as many berries as grandpa.

The blueberry shrubs can be represented as a string $s = s_1 s_2 \ldots s_N$ of length $N = 10^5$, consisting of characters `.` and `b`. If $s_i = \texttt{b}$, then there is a berry at position i, otherwise there is no berry there. There will be exactly $N/2$ berries to start with, and they are distributed uniformly at random.

The berry picking takes place in turns. In one move, a person can choose a position i ($1 \leq i \leq N - 3$), and pick all the berries at positions i, $i + 1$, $i + 2$, and $i + 3$. Grandpa makes the first move, then Erik makes a move, then grandpa, and so on. Grandpa will always greedily make a move that picks as many berries as possible. Among all such moves, he will pick one uniformly at random.

You are given the string s, and your task is to write a program that decides what moves Erik should make in order to pick at least as many berries as grandpa.

Interaction

The first line of input contains the number N, the length of the string s. This number will always be equal to 10^5, except for in the sample. Your program does not have to solve the sample to get accepted.

The second line contains the string s of length N. This string has exactly $N/2$ characters `b`, and their positions are generated uniformly at random.

Then, your program should start interacting with the grader. When it is grandpa's turn, you should read a single integer i ($1 \leq i \leq N - 3$), meaning that grandpa picks the berries at positions i, $i + 1$, $i + 2$, and $i + 3$. When it is your turn, you should instead print a number i, meaning that you pick the berries at positions i, $i + 1$, $i + 2$, and $i + 3$.

If it is your turn and there are no more berries, your program should terminate without printing anything more. If it is grandpa's turn and there are no berries, then the grader will not make any more moves, and your program should also terminate without printing anything more. If you picked at least as many berries as grandpa, you pass the test case.

Each turn, grandpa greedily makes a move that picks as many berries as possible, and among all such moves, he chooses one uniformly at random.

The strings s were generated beforehand, so they are the same across different submissions. However, grandpa's behaviour is randomized every submission, so it is possible to get different results if you submit the same code twice.

There will be 100 test cases. It is guaranteed that there exists a solution that passes this many cases with very high probability.

Example

In the sample, Erik and grandpa get 5 berries each, which means that Erik achieved his goal. At the very start of the game, the maximum number of berries possible to pick in one move is 3. There are a 4 possible choices of i that gets 3 berries: $1, 2, 3$ and 6. Grandpa chooses one of these uniformly at random which ends up being $i = 2$. After that, Erik chooses $i = 6$ which gives him three berries as well. In the last two turns, both grandpa and Erik pick two berries.

Read **Sample Interaction 1** **Write**

```
20
.bbb.b.bb...b.b...bb
2
```

```
6
```

```
13
```

```
17
```