

DMOPC '14 Contest 2 P2 - Cutting Logs

Time limit: 2.0s **Memory limit:** 64M

Recently, the Logging Company realized that some of the logs it owns may have defects in them. Being a reputable company, they cannot sell the defective portions of the logs. However, they don't want to waste their logs, so they will just cut out the defective portions and sell the remaining non-defective logs they get from the original log. Because the Company can sell longer logs for more money than many shorter logs, they want to make the remaining non-defective logs as long as possible. You have been hired to predict the results of this process.

Note: In case you were curious, the Logging Company can sell a log of length K units for K^2 units of money. You do not have to directly use this information in your program, but keep in mind that your goal is to maximize the sum of money earned from the remaining logs.

Input Specification

The first line will be the integer L ($1 \leq L \leq 100$), representing the length of the original log.

The second line will have L characters, the original log. A defective part will be represented by `X`, and a non-defective part will be represented by `O`.

Output Specification

The first line of output should be R , the number of resulting logs.

The next R lines of output each contain a log cut from the original log. **Assuming that the Logging Company cuts the log from left to right, you should output the resulting logs in the order that they are cut.** See the Sample Inputs for more details of this process.

Sample Input 1

```
4
OOXO
```

Sample Output 1

```
2
00
0
```

Explanation for Sample Output 1

The third part of the log needs to be cut out, and since we want the resulting logs to have the most value, having a log of length 2 and a log of length 1 is better than having 3 logs of length 1.

Sample Input 2

```
5
XXXXX
```

Sample Output 2

```
0
```

Explanation for Sample Output 2

Unfortunately, the whole log is defective.

Sample Input 3

```
7
X00X000
```

Sample Output 3

```
2
00
000
```

Sample Input 4

```
9
000X00X0X
```

Sample Output 4

```
3  
000  
00  
0
```

Sample Input 5

```
10  
0000XX0000
```

Sample Output 5

```
2  
0000  
0000
```