Arrow a Row

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Define an "arrow string" as a string that meets the following conditions:

- The length of the string is at least 5.
- The string starts with `>` and ends with `>>>`.
- The rest of the string consists only of `-`.

For example, `>-->>>` and `>--->>>` are valid arrow strings, while `>->>` and `>->->>>` are not.

Sauden gives you a string s of length n, consisting of `>` and `-`. You need to create s by performing a series of painting operations on a string of the same length n that consists entirely of `*`. In one painting operation, you can choose a substring of length at least 5 and transform it into an arrow string. The total number of operations you perform cannot exceed n.

If it is impossible to obtain the string s using no more than n painting operations, output `No`. Otherwise, output `Yes` and provide the details of the painting operations. If there are multiple solutions, output any.

Input

The first line contains a single integer T ($1 \le T \le 10^4$), indicating the number of test cases.

Each test case contains a string s of length n ($5 \le n \le 10^5$) in a single line, consisting only of `>` and `-`.

It is guaranteed that the sum of n over all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, if the given string cannot be obtained by performing no more than n painting operations, output `No` in a single line. Otherwise, output `Yes` and a positive integer m ($1 \le m \le n$), which denotes the number of painting operations to perform. Then output m lines, each contains two integers p ($1 \le p \le n-4$) and l ($5 \le l \le n+1-p$), indicating the starting position of the selected substring and its length.

Example

standard input	standard output
4 >>->>> >>>-> >>>>> >->>>>>>	Yes 2 1 5 2 5 No No Yes 2 2 7 1 5

Note

For the fourth test case in the example, the painting process is shown below::

******** → *>--->>> → >->>>>>>