Your task is to compress a string of no more than 200 characters, using the following scheme:

- adjacent repeats: $[S]k$

  which means: S repeated k times (where k is a one-byte integer. recall that the length of the string does not exceed 200)

- repeats with gaps: $[S]k\{S_1\}t_1\{S_2\}t_2 \ldots \{S_r\}t_r$, where $1 \leq t_i < k$, $t_i < t_{i+1}$

  which means: write S for k times, then insert string S_i after the t_i-th occurrence of S.

Note that the compressing is done recursively, so S, S_1, ..., S_r mentioned above can all be compressed further.

e.g. for the original string

```
I_am_WhatWhat_is_WhatWhat
```

the optimal compressed string is:

```
I_am_[What]4{_is_}2
```

Input

There are at most 20 test cases, each test case is a string containing no more than 200 printable characters, without whitespace characters (i.e., no spaces, no tabs), brackets (i.e. not in {'(',')','[',']','{','}'}) and digits.

Letters are case-sensitive.

Output

For each case, print the length of the minimal string, and a compressed string. Note that every one-byte integer should be counted as one character, even if it has two or three digits in its decimal form.

Sample Input

```
I_am_WhatWhat_is_WhatWhat
aaaabaaaaaaaabaaaaaaaabaaaa
??????????
```

Sample Output

```
19 I_am_[What]4{_is_}2
11 [[a]8{b}4]3
4 [?]10
```