3526 Sign Message Formatting

The *National Transportation Communications for ITS Protocol* (NTCIP) for communicating with highway signs with dynamic messages describes the messages using *Markup Language for Transportation Information* (MULTI). A MULTI string consists of text to be displayed together with embedded tags which describe formatting of the text and included dynamic elements. Tags begin with the open bracket ('`[`') character and end with the close bracket ('`]`') character. If an open bracket character is to appear in the text, it is represented as two open bracket characters. Similarly, if a close bracket character is to appear in the text, it is represented as two close bracket characters. This problem is concerned with formatting for character cell displays, which are rectangular arrays of character cells each of which can display a single character.

The tags to be supported for this problem are:

Tag	**Description**
`[nl]`	Start a new row of the array
`[sc`< *digit* >`]`	Insert < *digit* > blank character cells between each pair of text characters in the following string until changed.
`[/sc]`	Set inter-character spacing to zero (equivalent to `[sc0]`).
`[jl2]`	Set left justified text (the first character of the text is the leftmost character of the line).
`[jl3]`	Set center justified text (the number of character positions before and after the text on the line is the same or the number after is one more than the number before).
`[jl4]`	Set right justified text (the final character of the string is the rightmost character on the line).
`[jl5]`	Set fully justified text (an equal number of blank character spaces, as large as possible for the line length, is placed between each pair of characters in the text; the resulting string is centered in the line as for centered text).

For example, on a 24 character line (␣ indicates a blank character cell):

Format String	**Generated Output**
`[jl2]MESSAGE`	`MESSAGE␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣`
`[jl3]MESSAGE`	`␣␣␣␣␣␣␣␣MESSAGE␣␣␣␣␣␣␣␣␣`
`[jl4]MESSAGE`	`␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣MESSAGE`
`[jl2][sc2]MESSAGE`	`M␣␣E␣␣S␣␣S␣␣A␣␣G␣␣E␣␣␣␣␣`
`[jl5]MESSAGE`	`␣␣M␣␣E␣␣S␣␣S␣␣A␣␣G␣␣E␣␣␣`
`[jl2]THIS[jl3]IS A[jl4]MESSAGE`	`THIS␣␣␣␣␣␣IS␣A␣␣␣MESSAGE`

Letters within tags are case-insensitive. That is `[nl] = [NL] = [Nl] = [nL]`.

The default justification at the beginning of a message is left justification and the default character spacing is 0.

Justification and character spacing are maintained across `[nl]` tags.

The `[jl2]` and `[jl5]` tags may only be used before any text has been output on a line. Otherwise it is an error (`TAG CONFLICT`).

Once [jl5] text has been output on a line, no other justification tag may be set on that line. Otherwise it is an error (TAG CONFLICT).

The [jl3] tag may not be used after right justified text ([jl4]) has been output on a line. Otherwise it is an error (TAG CONFLICT).

A justification tag [jl?] with the same value as the current value does not cause a TAG CONFLICT error.

Extra character spacing specified by the [sc?] tag is ignored on lines with fully justified text. The full justification rules determine the extra spaces.

If too many characters are required on a line or too many lines are required in a message, it is an error (TOO BIG). A [nl] tag does not begin a new line unless followed by text output.

If left justified text and center justified text appear on the same line, there must be at least one blank character cell between the last character of left justified text and the first character of center justified text. Otherwise it is an error (TOO BIG).

If center justified text and right justified text appear on the same line, there must be at least one blank character cell between the last character of center justified text and the first character of right justified text. Otherwise it is an error (TOO BIG).

If left justified text and right justified text appear on the same line, there must be at least one blank character cell between the last character of left justified text and the first character of right justified text. Otherwise it is an error (TOO BIG).

The only tags allowed in a message are the seven tags listed above otherwise it is an error (BAD TAG). A malformed tag or an unmatched single open or closed bracket is a BAD TAG error.

For this problem you will write a program which takes as input the dimensions of the character cell array and a MULTI string and either outputs an error string or a correctly formatted message.

Input

The first line of input contains a single integer N, $(1 \leq N \leq 100)$, which is the number of datasets that follow. Each dataset consists of a single line containing an integer R, $(1 \leq R \leq 25)$, a blank, an integer C, $(1 \leq C \leq 80)$, a blank, and the remainder of the line is a ***MULTI-Text*** string. R is the number of rows in the character cell array, C is the number of columns in the character cell array, and the ***MULTI-Text*** is the text to be formatted.

Output

For each dataset, output the dataset number on a line by itself, followed by one of the error strings ('TAG CONFLICT', 'TOO BIG', 'BAD TAG') on a line by itself in the case of an error, or, R lines each of which has exactly C characters (other than terminating newlines) representing the formatted message using space characters for empty character cells.

The last line of output for a dataset result should be a single blank line.

Note: For ease in grading, a dataset that contains an error will only contain one type of error.

Note for Sample Output: (^ added after the last character of the lines for illustrative purposes. It should NOT appear in program output.)

Sample Input

```
7
4 24 [jl2]MESSAGE[nl][jl3]MESSAGE[nl][jl4]MESSAGE
2 24 This[jl3]is a[jl4]message
2 24 This is a very long message which will not fit
4 24 This[nl]message[nl]has[nl]too[nl]many[nl]lines
```

```
2 32 [jl3]This message has a [[ and a ]]
2 32 This is a bad tag[xy34]
2 32 [jl3]This message [jl5] has a tag conflict.
```

Sample Output

```
1
MESSAGE                 ^
        MESSAGE         ^
                 MESSAGE^
                        ^

2
This      is a   message^
                        ^

3
TOO BIG

4
TOO BIG

5
  This message has a [ and a ]  ^
                                ^

6
BAD TAG

7
TAG CONFLICT
```