PROBLEM H

THE "PAREN" THESIS

Jane Paren, was sitting around the house one day thinking about a multiplication machine she wanted to design. Her machine would take an ordered sequence of variables or numbers, like this:

$X_1,\ X_2,\ X_3,\ X_4,\ ?,\ X_N$

And then determine all possible ordered, multiplicative "groupings" of the objects like this:

$(X_1(X_2(X_3(X_4(\ldots(X_{N-1}*X_N)\ldots)))))$
: : :
: : :
$(((\ldots(((X_1*X_2)X_3)X_4)\ldots)X_{N-1})X_N)$

Jane hypothesized that a machine could produce these different groupings by repeatedly dividing the main sequence into two parts starting on the left and working toward the right, and then treating each of these sub−sequences just like the main sequence was treated. The very first division into two sub−sequences is shown here:

$((X_1)(X_2\ X_3\ \ldots\ X_{N-1}\ X_N))$

Later in the sequence another division of the main sequence into two sub−sequences is shown below:

$((X_1\ X_2\ X_3\ \ldots\ X_{j-1})(X_j\ \ldots X_{N-1}\ X_N))$

Both the left and right sequences must be subdivided from left to right just like the main sequence. You will help Jane Paren verify her hypothesis by writing a program that can group a sequence into a series of multiplications using parentheses. Your program will not perform the actual multiplications; it will only find and output all possible ordered parenthesized groupings in the order specified by Jane's hypothesis. file:///tmp/ACMContes "

Input

The input will consist of multiple lines of variable names or numbers.

Each variable name or number will be separated from its neighbors by spaces. The input line will be terminated by carriage−return/line−feed (standard end−of−line marker). Ignore multiple blank lines in the input and treat them as a single blank line.

The program will stop reading input when it reaches the end of the file.

Input for this problem will be from the text file **H.in**

Output

For each input line, the program will generate all possible multiplicative groupings, keeping the numbers or variables from the input line in their original order. **The order in which the multiplicative groupings are output does matter**!

Subsequences consisting of just one element will not have its' own set of parenthesis.

Subsequences consisting of just two elements are separated with the multiplication symbol * and have a set of parenthesis around them in the output, e.g. `(x*y)`.

For each element of the set of outputs calculated from a given input line, the output will display the line from which the group originated and the number within the group.

Outputs must conform to the same formatting as the sample output, including spaces!

Separate each set of output groupings by a blank line.

Sample I/O

Input:

```
1 2
North South East West
```

Output:

```
Input Line 1 Group 1. (1*2)

Input Line 2 Group 1. (North(South(East*West)))
Input Line 2 Group 2. (North((South*East)West))
Input Line 2 Group 3. ((North*South)(East*West))
Input Line 2 Group 4. ((North(South*East))West)
Input Line 2 Group 5. (((North*South)East)West)
```