1. Run-Length Encoding (RLE) is a simple form of data compression where consecutive elements of the data that are the same are replaced with a single value and the count of occurrences. In this problem, you are required to implement a C function that encodes a C-style string given as a parameter into its RLE form.

Constraints:

-   string will be at most 500 chars. (before and after compression)

Example:

| Before | After |
| --- | --- |
| "aaabbbccd" | "a3b3c2d1" |

2. You have a file containing text lines, each line possibly containing a specific pattern with square brackets denoting a range.

The pattern follows the format: <word>[<start>-<end>], where:

-   <word> represents a single word, consisting of alphanumeric characters (letters and digits).
-   <start> and <end> are positive integers, representing the inclusive range for the sequence.

Write a C function that reads each line from the file, identifies patterns matching the specified format (with a word as a prefix), and prints the expanded sequences for each line.

Example:

| File input | Function output |
| --- | --- |
| "This is an example[1-3] sentence." <br> "Another example[5-7] here." <br> "We have cat[1-3] here." <br> "Simple line." | "This is an example1, example2, example3 sentence." <br> "Another example5, example6, example7 here." <br> "We have cat1, cat2, cat3 here." <br> "Simple line." |

3. You are given a file containing lines, each representing a point in the XY plane. Each line consists of two floating-point numbers separated by a space, representing the coordinates {x, y} of a point.

Write a C function that takes the input file as a parameter, processes each line, and prints out the points in increasing order based on their x value and on decreasing order based on their y value.

Constraints:

- The length of each line will not exceed 100 characters.
- There may be an arbitrary number of points in the file.

Example:

| File input | Output |
| --- | --- |
| 1.0 2.5 | (1.0, 2.5) |
| 3.0 0 | (1.0, 0.5) |
| 1.0 0.5 | (2.0, 1.0) |
| 2.0 1.0 | (3.0, 0.0) |