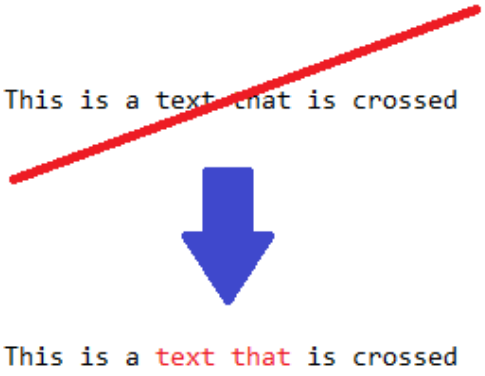
DSA Assignment #1



Intro

You are creating a browser engine that will be the fastest browser engine ever. To achieve this goal you decided to create some improvements to rendering algorithms. One of the improvements is to refresh only parts of the **text** when a line overlaps it. You decided to start your implementation with monospace fonts. Your primary task is to **detect words** affected by the line and **return these words in reverse order** (just a caprice). You even created a plan for your algorithm and implemented part of the code (you are given a **template java file**). So, you just have to implement the rest!

Plan

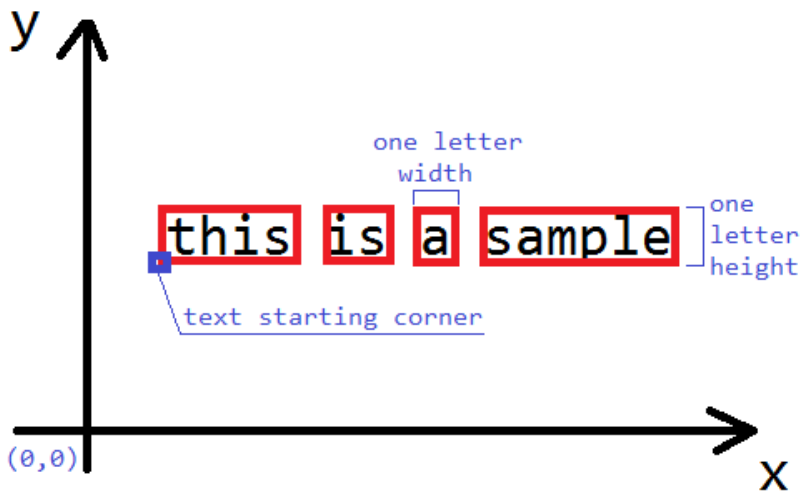
1) You decided to consider each **word as a rectangle** and use **points** to represent its corners.

```
public static class Point2D {
        ...
}

public static class Rectangle {
        ...
        public Rectangle(Point2D lowerLeft, Point2D upperTop, String tag) {
                ...
        }
}
```

2) And you already know, how to convert string to rectangles. The only thing – you want to use your **own list implementation** (because you know, how to do it better).

```
public static MyList getRectangles
        (String text, Point2D oneLetterSize, Point2D startCorner) {
        ...
}
```



3) Now you have to find which rectangles (and words) are overlapped by the line. Think about different cases of intersection. You would like to have this words in a reversed order, so you are using **your own stack** implementation as a return type.

```
public static MyStack getAffected
       (MyList rects, Point2D lineStart, Point2D lineFinish) {
       ...
}
```

4) Having a stack, it is very easy to create reverse order of words, just pop() them one by one.

NB: specific requirements are in the template file comments.

Input format explanation

File **browser.in** contains 2 lines. **First line** has a string that we will check for intersections. It has words separated by spaces.

The quick brown fox jumps over the lazy dog

Second line stores 8 real numbers in the following order:

```
textCornerX textCornerY letterSizeX letterSizeY lineStartX lineStartY lineEndX lineEndY
```

For example, second line: `0 0 0.5 1.1 0.1 -0.4 20 1.8`
Means that text lower left angle is [0, 0], letter size is [0.5, 1.1]. Line starts from [0.1, -0.4] and ends at [20, 1.8]

Output format explanation

Write all words that intersect with the line in reverse order separated by spaces into **browser.out** file.

Example

browser.in	browser.out
The quick brown fox jumps over the lazy dog 0 0 0.5 1.1 0.1 -0.4 20 1.8	over jumps fox brown quick