You have been tasked with finishing this implementation of TextEditor. The TextEditor should be able to print a string based on the input passed. The various operations of the TextEditor are listed below.

**Tips**

* Implement each operation one at a time and refactor your code as you go.
* Ignore the first array element ("0", "1", "2", etc) at first. It will be used in step 5, but is passed in to all test cases for consistency.
* Get as many test cases passing as you can in the 1hr time limit!
* There is no penalty for running your code. Run the tests early and often!
* **Your final score will be the maximum of your submitted scores, so submit early and often, don't wait until the last minute! If the clock runs out and you have not submitted yet, your score will be 0, even if you have partially passing tests.**

1. *INSERT* should append text:
2. input = [
3. ["0", "INSERT", "a"],
4. ["1", "INSERT", "b"],
5. ["2", "INSERT", "c"]
6. ]
7. // returns: "abc"
8. *DELETE* should remove the last character:
9. input = [
10. ["0", "INSERT", "abc"],
11. ["1", "DELETE"]
12. ]
13. // returns: "ab"

and

input = [

["0", "INSERT", "abc"],

["1", "DELETE"],

["2", "DELETE"]

]

// returns: "a"

*DELETE* should do nothing when there are no characters to delete:

input = [

["0", "INSERT", "a"],

["1", "DELETE"],

["2", "DELETE"]

]

// returns: ""

1. *UNDO* should undo the previous *INSERT* or *DELETE* operation:
2. input = [
3. ["0", "INSERT", "Hello"],
4. ["1", "INSERT", " World"],
5. ["2", "INSERT", "!"],
6. ["3", "UNDO"],
7. ["4", "UNDO"]
8. ]
9. // returns: "Hello"

and should do nothing when there are more *UNDO*s than other input:

input = [

["0", "INSERT", "Hello"],

["1", "UNDO"],

["2", "UNDO"]

]

// returns: ""

1. *REDO* should redo the previous *UNDO* operation:
2. input = [
3. ["0", "INSERT", "Hello"],
4. ["1", "INSERT", " World"],
5. ["2", "UNDO"],
6. ["3", "REDO"]
7. ]
8. // returns: "Hello World"

and should do nothing when there are more *REDO*s than *UNDO*s:

input = [

["0", "INSERT", "Hello"],

["1", "UNDO"],

["2", "REDO"],

["3", "REDO"]

]

// returns: "Hello"

and should only work immediately after an *UNDO* or *REDO* operation:

input = [

["0", "INSERT", "Hello"],

["1", "UNDO"],

["2", "INSERT", " World"],

["3", "REDO"]

]

// returns: " World"

1. input should be applied in chronological order according to the UNIX timestamp given.
2. input = [
3. ["1548185072722", "INSERT", "ola"],
4. ["1548185072721", "INSERT", "H"]
5. ]
6. // returns: "Hola"
7. *SELECT* should perform the operation following it on the range from start to end. Start is inclusive and end is exclusive (equivalently, you can think of start and end as the index of a cursor rather than the index of a character). If the selection is greater than length of text, select up to the end. If a *SELECT* follows another *SELECT*, the most recent should be used. If the start of the *SELECT* is outside the range of the text, it should be ignored. The different uses of *SELECT* are further described below.  
   *SELECT* and *DELETE* should remove the selected characters:

input = [

["1548185072721", "INSERT", "Hello"],

["1548185072722", "SELECT", "1", "3"],

["1548185072723", "DELETE"],

]

// returns: "Hlo"

*SELECT* and *INSERT* should replace the selected characters with the inserted characters:

input = [

["1548185072721", "INSERT", "Hello"],

["1548185072722", "SELECT", "1", "5"],

["1548185072723", "INSERT", "ola"],

]

// returns: "Hola"

*SELECT* and *BOLD* should insert \* characters before the first selected character and after the last selected character:

input = [

["1548185072721", "INSERT", "Hello"],

["1548185072722", "SELECT", "1", "3"],

["1548185072723", "BOLD"],

]

// returns: "H\*el\*lo"

Example

For input = [["0", "INSERT", "a"], ["1", "INSERT", "b"], ["2", "INSERT", "c"]], the output should be  
plaidTextEditor(input) = "abc".

Input/Output

* **[execution time limit] 3 seconds (java)**
* **[input] array.array.string input**

An array of operations need to be applied to the text editor. It is guaranteed that each operation is one of the one described in the description, all operation parameters are given in correct format, and that the text editor will never be in an incorrect state that is not described in the description.

* **[output] string**

The final text in the text editor after applying all operations.

**[Java] Syntax Tips**

// Prints help message to the console

// Returns a string

//

// Globals declared here will cause a compilation error,

// declare variables inside the function instead!

String helloWorld(String name) {

System.out.println("This prints to the console when you Run Tests");

return "Hello, " + name;

}