Robotique
CRC
Robotics

CRC

PROGRAMMING COMPETITION

PRELIMINARY PROBLEM 1

A FEW NOTES

- The complete rules are in section 5 of the rulebook.
- You have until **Friday, November 25, 11:59 pm** to submit your code
- Feel free to use the programming forum on the CRC discord to ask questions and discuss the problem with other teams. That's what it's there for!
- **We are giving you quick and easy-to-use template files for all languages allowed. Please use them.**

USING THE TEMPLATE FILE

- Basically, the template file uses your solve() code to transform a test case into its output and allows you to quickly check if it did what was expected or not. **All your code (except additional functions and classes, which should go right above it in the file) should be written in the solve() function.**
- **Do not touch anything lower than the solve().** There will be a list of input and a list of expected outputs, and a showing of how they would look in the problem statement for your convenience. There will also be a function that checks your answers by looping over all test cases of that input list. **Your solve() takes only one test case (always a String since it often represents a full line) at a time.**
- You can swap the **console's language to French** at the top of the file.

STRUCTURE

Every problem contains a small introduction like this about the basics of the problem and what is required to solve it. Points distribution is also given here for the preliminary problems.

Input and output specification:

In these two sections, we specify what the inputs will be and what form they will take, and we also say what outputs are required for the code to produce and in what format they shall be.

Sample input and output:

In these two sections, you will find a sample input (that often has multiple entries itself) in the sample input and what your program should produce for such an input in the sample output.

Explanation of the first output:

Sometimes, the problem might still be hard to understand after those sections, which is why there will also be a usually brief explanation of the logic that was used to reach the first output from the first input.

NO STRINGS ATTACHED

A lazy worker wrote a report on the AVIA operation a bit too quickly. He "forgot" to take into account multiple rules while writing the text, rules that you will have to correct to make sure the report is readable by the worker's superiors:

- Rule #1: The text should not have mathematical operations lurking in it, but only numbers instead. Therefore, 44-13 is not acceptable in a sentence and will have to be replaced by its result, 31. A quick reading by a software informs us that there will never be more than one mathematical operation in such mathematical statements. Also, all such mathematical operations are limited to the 4 basic ones, so +-*/. **The result of a division will be given as an integer, not as a real number. (6, not 6.0)**
- Rule #2: Every text sample to be adjusted will contain one or multiple sentences. The only problem is that all letters in said samples are written in lowercase. Every sentence's first letter will have to be put back to an uppercase state (including the first sentence). Two sentences will be separated by a terminal punctuation mark (.!?). Numbers at the beginning of a sentence should stay untouched.
- Rule #3: As there are no capital letters, there is also no way to know proper nouns from common ones. Here is a list of the proper nouns that should be considered and modified by your code in order to be given a capital letter at the start of the word:
  "Montreal", "Quebec", "Toronto", "Vancouver", "Canada", "Julie", "Jimmy", "Louis", "Andrae", "Francois", "Xavier", "Elrik", "Simon", "Jeff" and "Charles"

That's all for the normal rules that we usually take for granted while writing a text. However, our worker's superiors gave him additional instructions. Here are the two added rules that he forgot to implement:

- Rule #4: If a word is composed of an even number of letters, an underscore "_" should be added right in the middle (between its two central letters). For example, "maman" stays "maman" while "papa" becomes "pa_pa".
- Rule #5: If the word's total value is divisible by 5, the space following it should be replaced by a tabulation. A word's total value is calculated by adding all of the letters' individual values together. For the word's value. A is equal to 1, B is equal to 2, and so on up to Z, which is equal to 26. Hence, for example, "orange" gives a total of 60 and "violet" gives a total of 83. We would then write "orange	" and "violet". **We suggest that you look up how to insert a tabulation in a String for the sake of this rule. Also, all chars in programming have an assigned *ASCII* value. Looking up what it is, what it does and how to use it in your programming language could make a few things simpler for you here.**

It should be noted that rules #4 and #5 do not apply to all numbers and to all words starting with a capital letter. For the sake of these two rules, "d'abord" and "au-dessus" are each seen as a single word. "vraiment," is also a single complete word, the tabulation would then follow the punctuation sign at the end of the word if its value is divisible by 5, of

course. All of the following punctuation signs should be ignored when counting a word's letters or adding up its value (has a value of 0, adds 0 to the letters count): _-',.!? The text samples will not contain accents of any form.

Scoring:

Every rule applied well contributes 3 points to the team's score, up to 15 points for the whole problem. If rule #4 and/or rule #5 are applied to words starting with a capital letter and/or to numbers, that rule will only contribute half (so half of 3) of its points to the team's score.

Input specification:

Your code will have to take a small portion of text given as an input String (given like this to your solve() in the file). In the sample, a new line corresponds to a new input.

Output specification:

Your solve() function should return the adjusted and corrected String for that test case. In the sample, a new line corresponds to a new output. Tabulations won't render well here, but they will be taken well into account in the answers list in the template file.

Sample input:

```
dans l'ordre naturel des choses, 22+20 a toujours ete un nombre
important. peu importe le contexte, 42 represente la verite
vraie.
etant riche de 3*5 dollars, julie peut s'acheter les bonbons de
son choix! son ami jimmy, lui, possede 14-12 dollars et n'aura
donc pas ses bonbons desires. il devra se contenter d'acheter de
la gomme.
charles goes to his school every morning in montreal, quebec,
knowing that his dreams are about to be fulfilled.
what a beautiful day outside! it's almost like louis could touch
some grass! 20/5 kilos on his lawn, to be precise.
```

Sample output:

```
Dans l'or_dre naturel des cho_ses, 42 a touj_ours ete  u_n
nom_bre important. Peu importe l_e cont_exte, 42 repre_sente
l_a ver_ite vraie.
Etant riche d_e 15 dollars, Julie pe_ut s'ach_eter les bonbons
d_e son choix! Son ami Jimmy, lui, possede 2 dollars e_t
n'aura    do_nc pas ses bonbons desires. Il devra s_e contenter
d'ach_eter d_e l_a gomme.
Charles go_es t_o   his sch_ool every   morning   i_n Montreal,
Quebec, knowing th_at his dre_ams  are about t_o  b_e fulfilled.
```

```
What a beautiful day     outside! It's alm_ost    li_ke Louis
could     touch so_me grass! 4 kilos o_n his la_wn,    t_o  b_e
precise.
```

Explanation of the first output:

In the first case, “dans” and “peu” are both at the start of a sentence, meaning they become “Dans” and “Peu” in the output. “22+20” easily gives “42”. All words with an even count of letters like “l'ordre” get a “_” in the middle and become (ignoring the apostrophe) “l'or_dre”, for example. “Dans” doesn’t follow this rule, as there is a capital letter beginning the word. Both “42” are numbers, meaning this rule also doesn’t apply to them. There is a tabulation at the end of “ete”, “u_n”, “repre_sente” and “vraie.”, as all these words’ value is divisible by 5. In the second case, the word “Jimmy,” is not followed by a tabulation as rule #5 doesn’t apply to words starting with a capital letter even when they are in fact divisible by 5.