COS341 Academic Year 2023: **Optional Supplementary Practical**, related to **Chapter #7.**

In Section 7.3 of our book you read that *"an instruction-set description is a list of pairs where each pair consists of a pattern (a sequence of intermediate-language instructions) and a replacement (a sequence of machine-code instructions)"*.

In this **OSP** we want to ***simulate*** such a scenario by means of a few simplistic character-strings. Given for this purpose is now the following simplistic *matching table:*

*"**Intermediate** Code Pattern"*	*"**Target** Code Pattern"*
ACBA	**X**
BAB	**Y**
AC	**Z**

On these premises it is your task to design and implement a **recursive C++** function that fulfills *all* of the following **requirements**:

- The function must be implemented *purely* recursively (*no* While-loop and *no* For-loop);
- For input the function must take **reference(&) parameter** to a global string **s** of arbitrary length: *see the code snippet provided on the next page for illustration*;
- The function follows a ***greedy*** *strategy* of *"largest possible match first", with* ***backtracking*** to a previous choice-point in cases where an earlier greedy attempt does not lead to success;
- In so attempting to match the source patterns of above onto a given input string, the function
  - **returns a string** composed of the corresponding target patterns ***if*** the recursive matching attempt (possibly with backtracking) was eventually successful;
  - otherwise **returns** the string **"***mapping not possible***"** as an error message to indicate that no complete pattern-matching was possible on the given input string.

For example:

- On input string **s** = "ACACACBAACBAB" the function would return "**ZZXZY**", including a "greedy" attempt at the final "ACBA" before the final "B" (will not succeed, therefore backtrack, attempt "AC" instead, and finally succeed with the final "BAB");
- On input string **s** = "ACACACBAACBA" the function would return "**ZZXX**"
- On input string **s** = "BABACABAB" the function would return "mapping not possible"

Turn the page →

Your solution MUST be implemented by "filling in" the following given C++ Template:

```
// The C++ Testing Tool will be https://www.onlinegdb.com/online_c++_compiler

#include <iostream>
#include <cstring>
#include <iomanip>
#include <cstdlib>
#include <string>
#include <cctype> // NO other #include are allowed!
using namespace std; // NO other using are allowed!

// global
string s = "The input string will be inserted here by Prof.G.";
                    // NO other global variables are allowed!

// Here will be your Recursive Function implementation

int main()
{
    // Here may perhaps be some other stuff that you might need ...

    // Here will be the call to your Recursive Function ...

     // Here may perhaps be some other stuff that you might need ...

    cout << s; // Display the output string (after translation).
    // If the INPUT string cannot be translated then the output
    // string must contain the Error-Message.
    return 0;
}
```

Your solution MUST be submitted in an open-source ***.TXT** file: no other file formats are accepted**!**

And now

HAPPY CODING :)