SPECIFICATIONS

DASTRUC 1st MACHINE PROBLEM (MP1)

33% of the Final Project Grade

I. INTRODUCTION

This document contains the specifications of the 1st Machine Problem. **MAKE SURE THAT YOU UNDERSTAND AND FOLLOW THE SPECIFICATIONS PROPERLY. FAILURE TO DO SO WILL AUTOMATICALLY FAIL THIS MP.**

II. OBJECTIVES OF THE MP

a. To learn how to implement and apply the stack data structure in evaluating arithmetic, relational and logical expressions.

III. PROBLEM DESCRIPTION

I. INPUT

a. The input is an INFIX expression.
b. Operands are limited to positive integers (no need to consider negative numbers or floating-point numbers)
c. Operators include:
    i. Arithmetic operators: +, -, *, /, %, ^ (caret is for exponentiation)
    ii. Relational operators: >, <, >=, <=, !=, ==
    iii. Logical operators: !, &&, ||
d. Parentheses are used as grouping symbols.
e. Assumption: for simplicity let us assume that the INFIX expression is stored into a string with at most 256 characters. To simplify your task further, assume also that there are NO spaces separating the tokens.

Example input #1: 30+5

Example input #2: 30*(100 - 50) >999

II. OUTPUT

The required output are:

a. POSTFIX expression – there should be ONE blank space between two tokens
   For example input #1, the output should be: 30 5 +
   For example input #2, the output should be: 30 100 50 - * 999 >
b. Evaluated value of the postfix expression
   For example input #1, the evaluated value should be: 35
   For example input #2, the evaluated value should be: 1
   (Note: for relational and logical operators 1 means true, 0 means false)

IV. REQUIREMENTS

You are required to implement the following on your program.

1. Based on problem description above, as a developer you are responsible for adding error handling, usage of comments and usage of functions. Not adding those items would have deduction points.
2. The program is GUI based: 10 Best Python Libraries for GUI (2023) - Unite.AI.
   Take note: NO GUI, minus 50 pts and you must do some self-learning about Python GUI (there are multiple libraries for Python GUI).
3. Input should be place by the user in the GUI and the output should be shown in the GUI as well.

V. DELIVERABLES AND SUBMISSION DEADLINE

You need to submit **one (1)** item, namely:

a. Your Python script for the main script (filename should be <*lastname*>_main.py)

It is the student's responsibility to check the solution **(several times!)** for syntax and semantic errors before submission. Once submitted, no replacement will be accepted.

The source code should be **RECEIVED** via e-mail as a .zip file (along with the other MP files) before **June 19, 2023, 12:00 noon**. The recipient of your e-mail should be one of the following, depending on your DASTRUC professor. Make sure to CC your and your groupmate's e-mail addresses before sending the e-mail.

IT223, IT225 - John Raymound Javier: jljavier@nu-fairview.edu.ph

IT221, IT222 and IT224 - Jian Michael Wu: jowu@nu-fairview.edu.ph

The e-mail subject should be:

DASTRUC_<section>:
[space]<LASTNAME_MP1>[space]<FIRSTNAME_MP1>[space]MP1_<LASTNAME_MP2>[space]<FIRSTNAME_MP2>[space]MP2_<LASTNAME_MP3>[space]<FIRSTNAME_MP3>[space]MP3

where:

- LASTNAME_MP1 and FIRSTNAME_MP1 = the one who was assigned MP1
- LASTNAME_MP2 and FIRSTNAME_MP2 = the one who was assigned MP2
- LASTNAME_MP2 and FIRSTNAME_MP3 = the one who was assigned MP3
- You should replace [space] with " " (one space bar press)

Ex. DASTRUC_IT123: PEÑA_RYAN_MP1_CANLAS_JOMARY_MP2_LO_HANS_MP3

INCOMPLETE SUBMISSION and **NON-COMPLIANCE** with the instructions and specifications will automatically result in point deductions.

LATE SUBMISSIONS WILL BE ACCEPTED but with a deduction of ONE point for every minute or a fraction thereof in fairness to all students who submitted on time.

VI. CHECKING, TESTING, AND GRADING SCHEME

Your Python Scripts will be compiled and tested. Note that by default you will receive a perfect grade of 100% (for this MP only). It will remain as 100% if there are no warnings, syntax errors, and semantic errors; and that you complied properly with all the MP specifications. **A program with a syntax error will automatically be given a grade of ZERO (0).**

The contribution, in percentage, for each of the required functions are indicated in the table below:

Requirements	Percentage
GUI	20
Postfix Expression	30
Evaluated Value of the Postfix Expression	30
Error handling, Functions and Comments	20

** Please see your professor if you have any questions regarding this MP **