**Members of project:** Apoorva Singh, Pierre Rheeder

**Description:**

We will be creating a program that asks the user for an equation as input and then outputs the Reverse Polish Notation of that equation. The user will also has the option to ask for examples. In this case the program will display a list of equations along with their RPN format.

**Code Explanation:**

The code consists of three functions. The main() function contains an if statement used to check the user input correlating to what the user would like to do. The options are: to print examples, to let the user input their own equation, to ask for a project description or to quit the program.

The main function will call another function called tempchoice(), which asks for the user input.

The third function is the infix\_to\_rpn() function. This is the function which will convert an equation to Reverse Polish Notation. It takes a string type as the input and firstly prints that initial equation. The function also returns the RPN as string at the end. This function uses stacks, dictionaries and arrays to reorganise the equation into the correct notation. It will use a for loop to iterate through each character of the equation and correctly determine whether it should be added to the output string, or held in an array until further information is received.

**Grammar:**

G= {P, T, N, S} = {P, <operator>, (<alphabet>,<number>),(<expression>)}

Where Productions are

<expression> ::= <term> | <term> <expression>  
 <term> ::= <number> | <alphabet> | <operator> <expression>  
 <number> ::= [0-9]+ | [0-9]+.[0-9]+  
 <alphabet> ::= [a-z]+| [A-Z]+  
 <operator> ::= '+' | '-' | '\*' | '/' | '^' | '(' | ')'