```
//*************************************************************************************/
// Exercise 03: Infix2Postfix & Postfix Evaluator by Wu, Yi-Hung@ICE.CYCU       2020/09
//*************************************************************************************/
// This is the NO LIST version. You MUST rewrite it by using a linked list!
//*************************************************************************************/
#include <string>                                   // string class
#include <cstdlib>                                  // system, atoi
#include "DS1expSTACK.hpp"                          // header file for expSTACK class

#define MAX_LEN     255                             // maximum string length

using namespace std;

bool isExp();                   // check each symbol & remove useless ones
bool isNumber();                // check whether the character is a number
bool isOp();                    // check whether the character is an operator
bool isPar(const char);         // check whether the character is a parenthesis
bool isBalanced();              // check the validity of balanced parentheses
bool isInfix();                 // check whether it is an infix expression
bool infix2postfix();           // transform infix into postfix
void postfixEval();             // evaluate the value of postfix

int main(void)
{   int command = 0;                                // user command

    do
    {   string  infixS, postfixS;

        cout << endl << "* Arithmetic Expression Evaluator *";
        cout << endl << "* 0. QUIT                         *";
        cout << endl << "* 1. Infix2postfix Evaluation     *";
        cout << endl << "***********************************";
        cout << endl << "Input a choice(0, 1): ";
        cin >> command;                             // get a command
        cin.ignore(MAX_LEN, '\n');                  // clear the input buffer
        switch (command)
        {   case 0: break;
            case 1: cout << endl << "Input an infix expression: ";
```

```
                getline(cin, infixS);

// Mission One: check whether it is an infix expression
// isExp(), isNumber(), isOp(), isPar(), isBalanced(), isInfix()

// Mission Two: transform infix to postfix, stored as a linked list
// infix2postfix()

                break;
            default: cout << endl << "Command does not exist!" << endl;
        }   // end switch
    } while (command != 0);                     // '0': stop the program
    system("pause");                            // pause the display
    return 0;
}      // end main
```