C Lab (Part 2): Postfix Evaluator

Name:
Student ID:

Write a program that evaluates a postfix expression (assume it's valid) such as
6 2 + 5 * 8 4 / -

The program should read a postfix expression consisting of single digits and operators into a character array. Using stack functions, the program should scan the expression and evaluate it. The algorithm is as follows:

```
1) Append the null character ('\0') to the end of the postfix expression. When the null
   character is encountered, no further processing is necessary.
2) While '\0' has not been encountered, read the expression from left to right.
   If the current character is a digit,
        Push its integer value onto the stack (the integer value of a digit character is its
        value in the computer's character set minus the value of '0' in the computer's
        character set).
   Otherwise, if the current character is an operator,
        Pop the two top elements of the stack into variables x and y.
   Calculate y operator x.
   Push the result of the calculation onto the stack.
3) When the null character is encountered in the expression, pop the top value of the
   stack. This is the result of the postfix expression.
```

[*Note:* In 2) above, if the operator is '/', the top of the stack is 2, and the next element in the stack is 8, then pop 2 into x, pop 8 into y, evaluate 8 / 2, and push the result, 4, back on the stack. This note also applies to operator '-'.]

The arithmetic operations allowed in an expression are:

+ addition
- subtraction
* multiplication
/ division
^ exponentiation
% remainder

The stack should be maintained with the following declarations:

```
struct stackNode {
     int data;
     struct stackNode *nextPtr;
};
typedef struct stackNode StackNode;
typedef StackNode *StackNodePtr;
```

The program should consist of main and six other functions with the following function headers:

`int evaluatePostfixExpression( char *expr )`

Evaluate the postfix expression.

`int calculate( int op1, int op2, char operator )`

Evaluate the expression op1 operator op2.

`void push( StackNodePtr *topPtr, int value )`

Push a value on the stack.

`int pop( StackNodePtr *topPtr )`

Pop a value off the stack.

`int isEmpty( StackNodePtr topPtr )`

Determine if the stack is empty.

`void printStack( StackNodePtr topPtr )`

Print the stack.

The program should be able to give the following output:

```
Enter a postfix expression:
123*456^/7*-8*+
1       NULL
2       1       NULL
3       2       1       NULL
2       1       NULL
1       NULL
6       1       NULL
4       6       1       NULL
5       4       6       1       NULL
6       5       4       6       1       NULL
5       4       6       1       NULL
4       6       1       NULL
15625           4       6       1       NULL
4       6       1       NULL
6       1       NULL
0       6       1       NULL
7       0       6       1       NULL
0       6       1       NULL
6       1       NULL
0       6       1       NULL
6       1       NULL
1       NULL
6       1       NULL
8       6       1       NULL
6       1       NULL
1       NULL
48      1       NULL
1       NULL
NULL
49      NULL
The value of the expression is: 49
```

Submission Guidelines

a. Create a single file program. Name the file as “clab_p2.c”.
b. You need provide appropriate comments to make your code readable. (Note that if your code is not working and there are no comments, you may loss all the marks.)