Lab 9: Postfix Evaluation

Read a postfix expression from the given input file, and evaluate the value of the postfix expression using stack ADT.

1. Input

Obtain a postfix expression from the given input file (expr\_input.txt). The expression ends with #. A detailed specification of operators and operands is provided below.

Macintosh HD:private:var:folders:n2:r1pzhmbn34vg4rtp4fbzwg0w0000gn:T:Screenshot_3_28_15_3_46_PM.jpg

* Available operators: +, -, \*, /, and %
* Operands: single-digit numbers (1, 2, 3, 4, 5, 6, 7, 8, and 9)
* Conditions:
* The expression should be no more than 100 characters.
* The delimiter for the end of the expression is ‘#‘.
* No exception handling is required for checking whether the input file exists.

2. Data structure

struct Stack {

char \*key;

int top;

int max\_stack\_size;

};

3. Evaluation algorithm



* Iteratively obtain tokens until you meet the end of an expression.
* There are two rules for popping and pushing the operands from/to the stack:
* When you meet an operand (number), push it onto the stack.
* When you meet an operator, pop two operands from the stack and perform the operation, and push the result back to the stack.

4. Program description

* name: p6.c
* input: a postfix expression in a file
* output: evaluation result in the standard output

Submit to the gitlab your code. ( ~2020/5/21 23:59 )