Lab3-Flexible

Algorithm Explanation

1.First, get a character, which may be - + [ ], then select the correspoding subroutines.

2.IF do `pop` , we'd check the stack first, if the stack is empty , store a '_' to the array prepared for output. Otherwise, just move the pointer of stack i.e PointerL add 1, PointR substract 1.

3.IF do `push` ,just add the data to the location that correspoding pointer points, then move the pointer.

4. If get `Ente`r ,we add a x0000 to the end of the array, and just output with `Puts` .

Part of Code

select the corresponding subroutine

```
INPUT       AND R0, R0, #0
    GETC
    OUT
    ADD R4, R0, #-10
    BRz POPANS
    LD R4, MINUS
    NOT R4, R4
    ADD R4, R4, #1
    ADD R4, R4, R0
    BRz LEFTPOP
```

processing the input

```
LEFTPOP     ADD R1, R1, #0
    BRz EMPTY
    LDR R0, R2, #0
    STR R0, R3, #0
    ADD R3, R3, #1
    ADD R2, R2, #1
    ADD R1, R1, #-1
    BR INPUT
```

```
EMPTY       LD R0, UNDERLINE
    STR R0, R3, #0
    ADD R3, R3, #1
    BR INPUT
LEFTPUSH    AND R0, R0, #0
    GETC
    OUT
    STR R0, R2, #-1
    ADD R2, R2, #-1
    ADD R1, R1, #1
    BR INPUT
```

output

```
POPANS      AND R1, R1, #0
    LD R1, RESULT
OUTPUT      NOT R2, R1
    ADD R2, R2, #1
    ADD R2, R2, R3
    BRz END
    LDR R0, R1, #0
    OUT
    ADD R1, R1, #1
    BR OUTPUT
END         HALT
```

Q&A

Q: what' your data structure?
A: At first, the stack is empty, the pointerL and the pointerR points to the same one location. As long as there are one data in the stack, the two pointers move to seperate top one location. If there are need to pop , check the stack whether is empty , and do the subroutines.