Lab3:Flexible

Codes

Main part

```
INPUT   TRAP    x20
        TRAP    x21
        LD  R3,ENTER
        ADD R3,R3,R0
        BRz OUTPUT
        LD  R3,PLUS
        ADD R0,R0,R3
        BRzp F1
        JSR PUSHL
        BRnzp   INPUT
F1      BRp F2
        JSR POPL
        BRnzp   INPUT
F2      ADD R0,R0,R3
        ADD R0,R0,#-1
        BRp F3
        JSR PUSHR
        BRnzp   INPUT
F3      JSR POPR
        BRnzp   INPUT
OUTPUT  NOT R3,R4
        ADD R3,R3,#2
        LD  R4,RES
FINAL   ADD R0,R3,R4
        BRp FIN
        LDR R0,R4,#0
        TRAP    x21
        ADD R4,R4,#1
        BRnzp   FINAL

FIN     TRAP    x25
```

Push & Pop(Left)

```
PUSHL   TRAP    x20
        TRAP    x21
        STR R0,R1,#0
        ADD R1,R1,#-1
        RET
POPL    NOT R3,R1
        ADD R3,R3,R2
        BRp F4
        LD  R0,US
        STR R0,R4,#0
        ADD R4,R4,#1
        RET
F4      ADD R1,R1,#1
        LDR R0,R1,#0
        STR R0,R4,#0
        ADD R4,R4,#1
        RET
```

Only shows the important part.

Algorithm

We use to pointer to show the next address to insert a value,and another pointer to show the next output value to be stored.The other operate is the same as the queue operate.

Q&A

Q：If we keep push in right and pop in left, how can we keep the space is enough?
A: We can use the Wrap-Around queue to solve this problem.

Thoughts

The first time to achieve a data structure in assemble language, still many things to learn.