Report of lab3

孟澍

3210101819

2022 年 7 月 20 日

1 Algorithm explanation

This is the flowchart of my program.

Basic idea: Read the input, decide which operation need to be done, execute the operation. When the input is over, output.

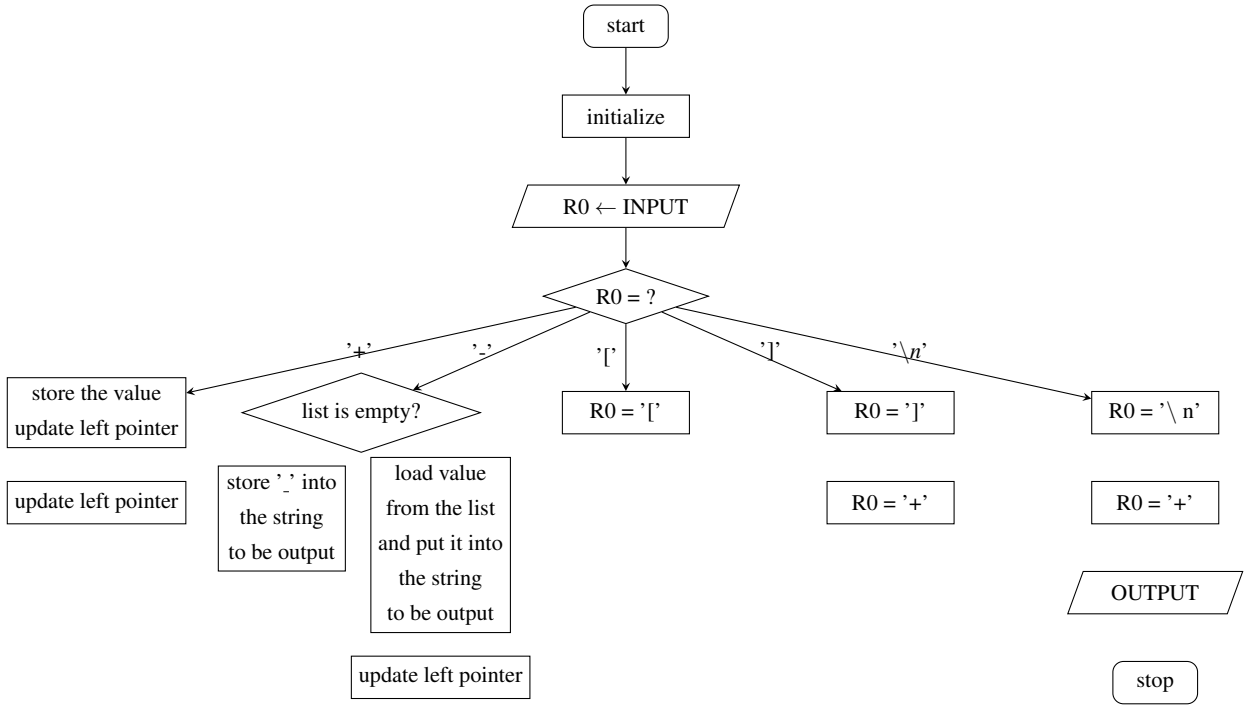

Fig 1: The flowchart of the program.

2 Source code

```
.ORIG       x3000
            AND R0, R0, #0     ; initialize
            LD R5, START       ; R5 is the left of the queue
            ADD R5, R5, #1     ; R5 <-- x4001
            LD R6, START       ; R6 is the right of the queue
```

```
                            ; R6 <-- x4000
                            ; Both R5 and R6 points to the
                            ; next position of the top
            LEA R4, OUTSTR  ; R4 is the address of the output string
;
INPUT       GETC
            OUT
            LD R1, PLUS
            NOT R1, R1
            ADD R1, R1, #1
            ADD R1, R0, R1
            BRnp #2
            JSR LPUSH       ; input char is '+'
            BRnzp CONT
            LD R1, MINUS
            NOT R1, R1
            ADD R1, R1, #1
            ADD R1, R0, R1
            BRnp #2
            JSR LPOP        ; input char is '-'
            BRnzp CONT
            LD R1, LBRACKET
            NOT R1, R1
            ADD R1, R1, #1
            ADD R1, R0, R1
            BRnp #2
            JSR RPUSH       ; input char is '['
            BRnzp CONT
            LD R1, RBRACKET
            NOT R1, R1
            ADD R1, R1, #1
            ADD R1, R0, R1
            BRnp #2
            JSR RPOP        ; input char is ']'
            BRnzp CONT
            LD R1, LF
            NOT R1, R1
            ADD R1, R1, #1
            ADD R1, R0, R1
            BRnp #2
            JSR OUTPUT      ; input char is '\n'
            HALT
CONT        BRnzp INPUT     ; get next char
;
LPUSH       LD R0, LLIMIT
            NOT R0, R0
            ADD R0, R0, #1
            ADD R0, R0, R5
            BRz FULL
            GETC
            OUT
            STR R0, R5, #0  ; store the value
            ADD R5, R5, #1  ; and update the left pointer
            RET
;
LPOP        NOT R0, R6
            ADD R0, R0, #1
            ADD R0, R0, R5
            ADD R0, R0, #-1
            BRz EMPTY
            ADD R5, R5, #-1 ; update the left pointer
            LDR R0, R5, #0
            STR R0, R4, #0  ; store the value to the string to be output
            ADD R4, R4, #1
            RET
;
RPUSH       LD R0, RLIMIT
            NOT R0, R0
            ADD R0, R0, #1
            ADD R0, R0, R6
            BRz FULL
            GETC
            OUT
            STR R0, R6, #0  ; store the value
            ADD R6, R6, #-1 ; and update the left pointer
```

```
81              RET
82  ;
83  RPOP        NOT R0, R6
84              ADD R0, R0, #1
85              ADD R0, R0, R5
86              ADD R0, R0, #-1
87              BRz EMPTY
88              ADD R6, R6, #1    ; update the right pointer
89              LDR R0, R6, #0
90              STR R0, R4, #0    ; store the value to the string to be output
91              ADD R4, R4, #1
92              RET
93  ;
94  OUTPUT      LEA R0, OUTSTR
95              PUTS
96              RET
97  ;
98  FULL        LEA R1, OVERFLOW
99              AND R0, R0, #0
100             ADD R0, R0, R1
101             PUTS
102             HALT
103 ;
104 EMPTY       LD R0, UNDERLINE
105             STR R0, R4, #0
106             ADD R4, R4, #1
107             RET
108 ;
109 PLUS        .FILL x002B
110 MINUS       .FILL x002D
111 LBRACKET    .FILL x005B
112 RBRACKET    .FILL x005D
113 LF          .FILL x000A
114 UNDERLINE   .FILL x005F
115 START       .FILL x4000
116 RLIMIT      .FILL x4100
117 LLIMIT      .FILL x3E00
118 OVERFLOW    .STRINGZ "ERROR!OVERFLOW!"
119 OUTSTR      .BLKW x0200
120 .END
```

3 Questions TA asked you and your answer in check

如何更好地利用内存空间？即：当只从左侧插入右侧取出时，如何保证内存空间的合理利用？