1.Algorithm

Deque implement by two stack grows in opposite directions. Store command in comm[], then traverse them,each time detect op +-[],then move corresponding pointer and store data when meet push operand

```
    char a[201];char * p =a;int lefttop = 100 ,righttop = 101; char
command[110];
    int commcnt=0; while((temchar = getchar())!= '\n') command[comm++]=temchar;
// for INPUT
    while(comm[j] is not empty) // HANDLE COMM
        char op = command[j]; char operand = command[j+1]; // j is command
pointer
        switch (op) {
        case '+': j+=2;
            a[lefttop--] = operand;  // use char op and char operand
        case '-': j++;
            if(righttop-lefttop!=1) putchar( a[++lefttop] );    else
putchar('_'); // empty
        case '[': j+=2;
            a[righttop++] = operand;
        case ']': j++;
            if(righttop-lefttop!=1) putchar( a[--righttop]);    else
putchar('_');
        }
```

2.Essential parts of code with sufficient comments

```
;INPUT while((temchar=getchar()!='\N') comm[cnt++]=temchar
INPUT   TRAP X20 ;GETCHAR
TRAP X21 ;PUTCHAR
LD  R3,NEGLF
ADD R3,R0,R3
BRZ INPUTOUT    ; if Temchar == '\n' input finish
STR R0,R4,#0    ;COMM[CNT] = TEMCHAR
ADD R4,R4,#1    ;P=&COMM,P++
ADD R5,R5,#1    ;CNT++
BR  INPUT
INPUTOUT LD R4,COMMPOS
WHILE
    ADD R5,R5,#-1    ;WHILE(CNT--)
    BRN DONE
    LDR R0,R4,#0    ;R0=OP
    LD  R3,NEG_LEFT_PUSH // Detect 4 operation signal
    ADD R3,R0,R3    ;OP == +
    BRZ LEFT_PUSH
    ... ; - [ ] is same as detecting '+'
LEFT_PUSH
    LDR R0,R4,#1    ;R0 = OP
```

```
    ADD R4,R4,#2    ;J+=2 means we use 2 characters in comm[]
    STR R0,R1,#0    ;A[LEFTTOP] = OPERAND
    ADD R1,R1,#-1   ;LEFTTOP--
    BR  WHILE

LEFT_POP
    ADD R4,R4,#1
    ADD R3,R1,#0
    NOT R3,R3
    ADD R3,R2,R3     ;RIGHTTOP - LEFTTOP == 1 check whether EMPTY
    BRZ POPFAIL     ;if so then we print '_'
    ADD R1,R1,#1    ;LEFTTOP++
    LDR R0,R1,#0    ;
    TRAP X21        ;PUTCHAR(A[LEFTTOP])
    BR WHILE
RIGHT_PUSH  ...     ;same as LEFT except the RIGHTTOP++ when PUSH and RIGHTTOP--
when POP ,opposite to LEFT
RIGHT_POP   ...     ;and EMPTY CONDITION is also RIGHTTOP - LEFTTOP == 1
POPFAIL
    LD  R0,DASH
    TRAP X21        ;PUTCHAR('_')
    BR  WHILE
DONE
HALT
DASH            .FILL X005F ; '_'
NEG_LEFT_PUSH   .FILL xFFD5 ; -'+'
NEG_LEFT_POP    .FILL xFFD3 ; -'-'
NEG_RIGHT_PUSH  .FILL xFFA5 ; -'['
NEG_RIGHT_POP   .FILL xFFA3 ; -']'
NEGLF           .FILL xFFF6 ; '\n'
COMMPOS .FILL   COMM        ; store command's address
BIAS    .FILL   100         ; move LEFTTOP to A[100]
DEQUE   .BLKW   201         ; store data
COMM    .BLKW   200         ; store command
.END
```

3.Questions

1.Explain your constant used

NEGLF detect '\n' in input phase , COMM store input command string,deque store data pushed. NEG_LEFT_PUSH NEG_LEFT_POP NEG_RIGHT_PUSH NEG_RIGHT_POP detect '+' '-' '[' ']' BIAS to move leftpointer to A[100] ,DASH for output '_' when empty

2.If input string length larger than 100,what will happen

Since COMM array is inserted and deleted through CNT++, it is not a problem to read and store the length of more than 100, but the leftstack may crash, because Deque does not handle overflow situation, so if you keep using LEFTPUSH, the reserved space will be insufficient,which will cause the program area to be overwritten. The solution is to set up a sentinel, and when the lefttop move to the sentinel position, it is sent to the rightmost sentinel, similar to the practice of circular queue. The condition of FULL is LEFTTOP-RIGHTTOP==1