Lab3 Report

1. Algorithm explanation

First, we need to read the operations and operands from the console and store it. Because we have to echo after input, so we can't just read an operation and process it. Because we can't output the result until the input is over. To echo, we just need to out right after getc, then we can output the character, including the ENTER, because when we read ENTER, it means it's time to change to next line to prepare to outputting. Additionally, we need to check whether we get an ENTER to determine to continue the read procedure.

Second, we need to construct the list according to the operations. We use r4 to point at the left end of the list and we let r4 keep track of the leftist number. We use r5 to point at the right end of the list and we let r5 keep track of the rightest number. Initially, we define r4=r5=x8000, and we will make the #1000 address near x8000 to x0000 by blkw. We use r3 to point at the operations we are now processing with.

Third, if we read an + or [from *r3, we need to mov r3 to the next character to get the operand and store it in the list. If the list is empty, which means that r4 = r5 and *r4 = 0 , then we just store the char in r4 without moving r4 or r5. If the list is not empty, then we just need to decrease r4 and store the char in r4 or increase r5 and store the char in r5. If we read a – or] from *r3 , then we need to check if the list is empty ,which means r4 = r5 and *r4 = 0 , if the list is empty , then we need to output '_' . If the list is not empty then we just output the number in r4 and cover it with 0 and increase r4 or output the number in r5 and cover it with 0 and decrease r5. If *r3 is 0 then we have done, we need to halt.

2. Essential codes

```
read    getc
        out
```

In read part we out put right after getc regardless of whether it I ENTER or not.

```
        lea r3,oper
output  ldr r0,r3,#0;get the opration
        brz done    ;if it's x00 then we have done
        add r3,r3,1
        ld r1,negmin ;to indentify which operaion it is
        add r1,r1,r0
        brz charmi
        ld r1,negadd
        add r1,r1,r0
        brz charad
        ld r1,negle
        add r1,r1,r0
        brz charle
;if it is ]
charri  ldr r0,r5,#0
```

Here we are judging which operation it is , if it is not - + and [then it must be] , so right

after the judging part is the] part.

```
;initialize the memory space for the list
.orig x7D00
.blkw 1000
.end
```

3. TA questions

Q: How many elements can be put in this list?

A: Because I don’ t use the stack, so we can store from the end of my code segment to XFDFF. But in that case, I should initialize more addresses to 0.

Q: What if I want to add x8000 chars to the left, can your structure handle all these data?

A: I need to use wrap, if r4 move to the left bound of the list, we need to check if the right part of the space is empty or not, if it is empty now, then we can move r4 to the right boundary and keep storing data. If we use wrap, then the list is full if we find r4 will be equal to r5 after the push, which means r4 has covered a whole lap to be equal to r5 and therefore used up all the space.