1. 6.16 An LC-3 program is located in memory locations x3000 to x3006. It starts executing at x3000. If we keep track of all values loaded into the MAR as the program executes, we will get a sequence that starts as follows. Such a sequence of values is referred to as a trace.

   We have shown below some of the bits stored in locations x3000 to x3006. Your job is to fill in each blank space with a 0 or a 1, as appropriate.

MAR Trace
x3000
x3005
x3001
x3002
x3006
x4001
x3003
x0021

x3000	0	0	1	0	0	0	0									
x3001	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1
x3002	1	0	1	1	0	0	0									
x3003																
x3004	1	1	1	1	0	0	0	0	0	0	1	0	0	1	0	1
x3005	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
x3006																

简答题 (4 分) 4 分

x3000 0010 000 0 0000 0100
x3002 0001 0000 0010 0001
x3002 1011 000 0 0000 0011
x3003 1111 0000 0010 0001
x3004 1111 0000 0010 0101
x3005 0000 0000 0011 0000
x3006 0100 0000 0000 0001

2. 7.4 Create the symbol table entries generated by the assembler when translating the following routine into machine code:

```
        .ORIG    x301C
        ST  R3, SAVE3
        ST  R2, SAVE2
        AND R2, R2, #0
    TEST    IN
        BRz TEST
        ADD R1, R0, #-10
        BRn FINISH
        ADD R1, R0, #-15
        NOT R1, R1
        BRn FINISH
        HALT
    FINISH  ADD R2, R2, #1
        HALT
    SAVE3   .FILL   x0000
    SAVE2   .FILL   x0000
        .END
```

简答题 (2 分) 2 分

附件名
大小

7y.4.jpg

3. 7.8 An engineer is in the process of debugging a program she has written. She is looking at the following segment of the program and decides to place a breakpoint in memory at location 0xA404. Starting with the PC = 0xA400, she initializes all the registers to zero and runs the program until the breakpoint is encountered.

   Show the contents of the register file (in hexadecimal) when the breakpoint is encountered.

```
xA400   THIS1   LEA R0, THIS1
xA401   THIS2   LD  R1, THIS2
xA402   THIS3   LDI R2, THIS5
xA403   THIS4   LDR R3, R0, #2
xA404   THIS5   .FILL   xA400

```

简答题 (4 分) 4 分

show in figure

附件名
大小

7.8.JPG

4. 7.10 The following program fragment has an error in it. Identify the error and explain how to fix it.

   Will this error be detected when this code is assembled or when this code is run on the LC-3?

```
1         ADD R3, R3, #30
2         ST  R3, A
3         HALT
4     A   .FILL   #0
```

简答题 (2 分) 2 分

the instruction "ADD R3, R3, #30" contains an immediate value that is too large to be stored. the instruction can't be translated and run ,thus the error will be detected ,and not run.

5. 7.18 The following LC-3 program compares two character strings of the same length. The source strings are in the .STRINGZ form. The first string starts at memory location x4000, and the second string starts at memory location x4100. If the strings are the same, the program terminates with the value 1 in R5. Insert instructions at (a), (b), and (c) that will complete the program.

```
1         .ORIG x3000
2         LD  R1, FIRST
3         LD  R2, SECOND
4         AND R0, R0, #0
5     LOOP    -------------- (a)
6         LDR R4, R2, #0
7         BRz NEXT
8         ADD R1, R1, #1
9         ADD R2, R2, #1
10        -------------- (b)
11        -------------- (c)
12        ADD R3, R3, R4
13        BRz LOOP
14        AND R5, R5, #0
15        BRnzp   DONE
16    NEXT    AND R5, R5, #0
17        ADD R5, R5, #1
18    DONE    TRAP    x25
19    FIRST   .FILL x4000
20    SECOND  .FILL x4100
21        .END
22
```

简答题 (3 分) 3 分

a: LDR R3,R1,#0
b: NOT R3,R3
c: ADD R3,R3,#1

6. 7.23 The following LC-3 program determines whether a character string is a palindrome or not. A palindrome is a string that reads the same backwards as forwards. For example, the string “racecar” is a palindrome. Suppose a string starts at memory location x4000 and is in the .STRINGZ format. If the string is a palindrome, the program terminates with the value 1 in R5. If not, the program terminates with the value 0 in R5. Insert instructions at (a)–(e) that will complete the program.

```
1         .ORIG   x3000
2         LD  R0, PTR
3         ADD R1, R0, #0
4      AGAIN   LDR R2, R1, #0
5         BRz CONT
6         ADD R1, R1, #1
7         BRnzp   AGAIN
8      CONT    ------------- (a)
9      LOOP    LDR R3, R0, #0
10        ------------- (b)
11        NOT R4, R4
12        ADD R4, R4, #1
13        ADD R3, R3, R4
14        BRnp    NO
15        ------------- (c)
16        ------------- (d)
17        NOT R2, R0
18        ADD R2, R2, #1
19        ADD R2, R1, R2
20        BRnz    YES
21        ------------- (e)
22     YES AND R5, R5, #0
23        ADD R5, R5, #1
24        BRnzp   DONE
25     NO  AND R5, R5, #0
```

简答题 (5 分)　5 分

A. ADD R1,R1,#-1
B. LDR R4,R1,#0
C. ADD R0,R0,#1
D. ADD R1,R1.#-1
E. BR LOOP