

HW06_solution

1.The following program is supposed to print the number 5 on the screen. It does not work. Why? Answer in no more than ten words, please.

```
1      .ORIG  x3000
2      JSR   A
3      OUT          ;TRAP x21
4      BRnzp DONE
5  A    AND   R0,R0,#0
6      ADD   R0,R0,#5
7      JSR   B
8      RET
9  DONE HALT
10     ASCII .FILL x0030
11     B    LD   R1,ASCII
12     ADD   R0,R0,R1
13     RET
14     .END
```

Need to save R7 so 1st service routine can return. Second RET overwrites the first RET value.

2.The following LC-3 program is assembled and then executed. There are no assemble time or run-time errors. What is the output of this program? Assume all registers are initialized to 0 before the program executes.

```
1      .ORIG x3000
2      ST R0, #6 ; x3007
3      LEA R0, LABEL
4      TRAP x22
5      TRAP x25
6  LABEL .STRINGZ "FUNKY"
7  LABEL2 .STRINGZ "HELLO WORLD"
8      .END
```

FUN

3.The following nonsense program is assembled and executed.

```

1      .ORIG x4000
2      LD R2,BOBO
3      LD R3,SAM
4  AGAIN  ADD R3,R3,R2
5      ADD R2,R2,#-1
6      BRnzp SAM
7  BOBO   .STRINGZ "Why are you asking me this?"
8  SAM    BRnp AGAIN
9      TRAP x25
10     .BLKW 5
11  JOE   .FILL x7777
12     .END

```

How many times is the loop executed? When the program halts, what is the value in R3? (If you do not want to do the arithmetic, it is okay to answer this with a mathematical expression.)

Work: BOBO is length 28 (27 + 1 for null). BRnp AGAIN in binary is 0000 101 #-32 = 0000 101 1 1110 0000 = x0BE0. R3 holds x0BE0. R2 starts with the value of W which is x57. R. The loop executes 57 times. The final value of R3 is $x0BE0 + (x57 + x1) * x57 / x2 = x0BE0 + x0EF4 = x1AD4$ or #6868. Note that x0BE0 is #3040.

4.The program below, when complete, should print the following to the monitor:

ABCFGH

Insert instructions at (a)-(d) that will complete the program.

```

1      .ORIG x3000
2      LEA R1, TESTOUT
3  BACK_1 LDR R0, R1, #0
4      BRz NEXT_1
5      TRAP x21
6      ADD R1, R1, #1 ;(a)
7      BRnzp BACK_1
8      ;
9  NEXT_1 LEA R1, TESTOUT
10 BACK_2 LDR R0, R1, #0
11      BRz NEXT_2
12      JSR SUB_1
13      ADD R1, R1, #1
14      BRnzp BACK_2
15      ;
16 NEXT_2 HALT ;(b)
17      ;
18 SUB_1 ADD R0, R0, #5 ;(c)

```

```

19 K      LDI    R2, DSR
20      BRzp   K          ; (d)
21      STI    R0, DDR
22      RET
23 DSR    .FILL  xFE04
24 DDR    .FILL  xFE06
25 TESTOUT .STRINGZ "ABC"
26      .END

```

5. Shown below is a partially constructed program. The program asks the user his/her name and stores the sentence "Hello, name" as a string starting from the memory location indicated by the symbol HELLO. The program then outputs that sentence to the screen. The program assumes that the user has finished entering his/her name when he/she presses the Enter key, whose ASCII code is x0A. The name is restricted to be not more than 25 characters.

Assuming that the user enters Onur followed by a carriage return when prompted to enter his/her name, the output of the program looks exactly like:

Please enter your name: Onur

Hello, Onur

Insert instructions at (a)-(d) that will complete the program.

```

1      .ORIG  x3000
2      LEA    R1, HELLO
3 AGAIN LDR    R2, R1, #0
4      BRz    NEXT
5      ADD    R1, R1, #1
6      BR    AGAIN
7 NEXT  LEA    R0, PROMPT
8      TRAP   x22          ; PUTS
9      LD    R3  NEGENTER   ; a
10 AGAIN2 TRAP  x20        ; GETC
11      TRAP  x21        ; OUT
12      ADD    R2, R0, R3
13      BRz    CONT
14      STR    R0, R1, #0   ; b
15      ADD    R1, R1, #1   ; c
16      BR    AGAIN2
17 CONT  AND    R2, R2, #0
18      STR    R2, R1, #1   ; d
19      LEA    R0, HELLO
20      TRAP   x22          ; PUTS
21      TRAP   x25          ; HALT
22 NEGENTER .FILL  xFFF6    ; -x0A
23 PROMPT  .STRINGZ "Please enter your name: "
24 HELLO   .STRINGZ "Hello, "

```

25

.BLKW #25

26

.END