Problem 1

Suppose the following LC-3 subroutine implements a new service routine called GETS. The subroutine will store the input string starting at the address in R0 and then return to normal execution. It performs this operation by repeatedly taking input characters from the keyboard and storing it in the location specified by R0 until it sees the '\n' character.

a. Fill in the missing instructions

```
.ORIG x0650
ST R0, R0_TMP
ST R1, R1_TMP
ST R2, R2_TMP
L1 LDI R1, KBSR
       _____ R2, KBDR ; Load value in the KBDR into R2
LD R1, NEGCHAR
ADD R1, R1, R2
BRz DONE ; Check for '\n'
STR R2, R0, #0
ADD R0, R0, #1
BRnzp L1
DONE _____
STR R2, R0, #0 ; Store NULL CHAR
LD R2, R2_TMP
LD R1, R1_TMP
LD R0, R0_TMP
```

b. Assume the above assembly code is a service routine that can be called using TRAP x55. What is the address of the corresponding System Control Block entry and what are its contents? Give your answer in hex.

<table>
<thead>
<tr>
<th>Address of Trap Vector Table Entry</th>
<th>Contents at this locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBSR .FILL xFE00 ; Address of KBSR</td>
<td></td>
</tr>
<tr>
<td>KBDR .FILL xFE02 ; Address of KBDR</td>
<td></td>
</tr>
<tr>
<td>NEGCHAR .FILL xFF6 ; Negative value of character '\n'</td>
<td></td>
</tr>
<tr>
<td>R0_TMP .FILL 0</td>
<td></td>
</tr>
<tr>
<td>R1_TMP .FILL 0</td>
<td></td>
</tr>
<tr>
<td>R2_TMP .FILL 0</td>
<td></td>
</tr>
<tr>
<td>.END</td>
<td></td>
</tr>
</tbody>
</table>

c. Describe briefly what happens during the execution of the instruction TRAP x55.
Problem 2
What is the problem with the below definition of a subroutine called PUTCH? How will you fix it?

```
.ORIG x3010
PUTCH
    ST R0, TMP_R0
    ADD R0, R4, 0
    OUT ; TRAP x21 displays the character in R0
    LD R0, TMP_R0
    RET
TMP_R0 .FILL 0
.END
```

Problem 3
What do the following programs do?

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
</tr>
</thead>
</table>
| .ORIG x3000  
LD R0, CHAR  
LD R1, LAB  
ADD R1, R1, #1  
ST R1, LAB  
LAB GETC  
HALT  
CHAR .FILL x72  
.END | .ORIG x4000  
LD R2, LOW_A  
NOT R2, R2  
ADD R2, R2, #1  
LEA R0, STRG  
L1 LDR R1, R0, #0  
BRz DONE  
ADD R3, R1, R2  
BRnp SKIP  
LD R1, UPP_A  
STR R1, R0, #0  
SKIP ADD R0, R0, #1  
BRnzp L1  
DONE LEA R0, STRG  
PUTS  
HALT  
LOW_A .FILL x61 ; ASCII Character 'a'  
STRG .STRINGZ "Salt and Pepper"  
UPP_A .FILL x41 ; ASCII Character 'A'  
.END |

Problem 4
(a) If a service routine is defined at address x0320 and the contents of memory location x42 is x0320, how would you invoke the service routine?
(b) What would you need to do to make the keyboard work in interrupt mode?
(c) What will happen if the KBSR is not checked before reading the KBDR?
(d) What will happen if DSR is not checked before writing into DDR?
(e) Why do most devices use asynchronous I/O?
(f) What is the advantage of polling over interrupt and vice versa?