1) What is the output of the following program?

```
         .ORIG x3000
          LD       R1, LETA
          LEA      R0, BUFFER
          STR      R1, R0, #0
          ADD      R1, R1, #1
           STR      R1, R0, #1
          ADD      R1, R1, #1
           STR      R1, R0, #2
          ADD      R1, R1, #1
           STR      R1, R0, #3
           TRAP     x22
           LD       R0, LF
           TRAP     x21
           LEA      R0, STRING
           TRAP     x22
           TRAP     x25
LF      .FILL     x000A
LETA    .FILL     x0041
BUFFER .BLKW     #4
STRING .STRINGZ "EFGH"
          .END
```

ABCDEFGH
EFGH

2) What does the following do?

```
          LEA    R3,Binary
          LD     R6,ASCII
          LD     R7,COUNT
AGAIN     TRAP   x23
          ADD    R0,R0,R6
          STR    R0,R3,#0
          ADD    R3,R3,#1
          ADD    R7,R7,#-1
          BRp    AGAIN
          BRnzp  NEXT_TASK
ASCII     .FILL  xFFD0
COUNT     .FILL  #10
Binary    .BLKW  #10
```

Reads in 10 digits and stores them in Binary format at Binary

3) What does the following code do?

```
START    LDI      R1, KBSR
         BRzp     START
         LDI      R0, KBDR

EO       LDI      R1, DSR
         BRzp     EO
         STI      R0, DDR

         BRnzp    NEXT_TASK
KBSR    .FILL    xFE00
KBDR    .FILL    xFE02
DSR     .FILL    xFE04
DDR     .FILL    xFE06
```

Echo character typed to the display

4) What does the following program do?

```
              .ORIG x3000
              AND R5, R5, #0
               AND R3, R3, #0
               ADD R3, R3, #8
               LEA R0, BB
               LDR R1, R0, #1
               LDR R1, R1, #0
               ADD R2, R1, #0
AGAIN         ADD R2, R2, R2
               ADD R3, R3, #-1
               BRp AGAIN
               LDR R4, R0, #0
               AND R1, R1, R4
               NOT R1, R1
               ADD R1, R1, #1
               ADD R2, R2, R1
               BRnp  NO
               ADD R5, R5, #1
NO            TRAP x25
BB            .FILL xFF00
               .FILL x4000
               .END
```

Puts 1 in R5 if the high byte of the data in location x4000 is equal to the low byte of the data in location x4000.

5) What does the following code do?

```
               .ORIG    x0450
               ST       R7, SaveR7
               ST       R0, SaveR0
               ST       R1, SaveR1
               ST       R3, SaveR3
Loop           LDR      R1, R0, #0
               BRz      Return
L2             LDI      R3,DSR
               BRzp     L2
               STI      R1, DDR
               ADD      R0, R0, #1
               BRnzp    Loop
Return         LD       R3, SaveR3
               LD       R1, SaveR1
               LD       R0, SaveR0
               LD       R7, SaveR7
               RET
DSR            .FILL    xFE04
DDR            .FILL    xFE06
SaveR0         .FILL    x0000
SaveR1         .FILL    x0000
SaveR3         .FILL    x0000
SaveR7         .FILL    x0000
               .END
```

puts.asm: This service routine writes a NULL-terminated string to the console. It services the PUTS service call (TRAP x22). Inputs: R0 is a pointer to the string to print. Context Information: R0, R1, and R3 are saved, and R7 is lost in the jump to this routine