1. Implement the following high-level CASE statement using LC-2 assembly code. Do not use self-modifying code as shown in class.

```assembly
Answer
.orig x3000
loop lea r0, prompt ;prompt "Enter a number?"
    trap x22
    trap x20 ;get ascii number
    trap x21 ;echo number
    ld r1, asciidi ;convert from ascii number to decimal
    add r1, r0, r1
    add r1, r1, r1 ;multiply by 2 because each case uses 2 bytes
    ld r0, lf ;new line
    trap x21
    lea r7, case0 ;get base address of cases
    add r7, r1, r1 ;r7 contains address of correct case
    ret ;jump to case

case0 lea r0, c0str
    jmp display

case1 lea r0, c1str
    jmp display

case2 lea r0, c2str
    jmp display
display trap x22
    ld r0, lf
    trap x21
```
2. In the Tic-Tac-Toe game, we need to check for three adjacent X's to determine if player X has won. Write a subroutine to perform this check. The subroutine returns with the "P" (positive) flag set if X has won, and reset otherwise. The 3x3 Tic-Tac-Toe board is stored in 9 consecutive memory locations as defined by the statement and picture:

```
Board .BLKW 9, x0000

<table>
<thead>
<tr>
<th>location 0</th>
<th>location 1</th>
<th>location 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>location 3</td>
<td>location 4</td>
<td>location 5</td>
</tr>
<tr>
<td>location 6</td>
<td>location 7</td>
<td>location 8</td>
</tr>
</tbody>
</table>
```

The contents of these locations are defined as follows:
- 0 = empty
- 1 = X in location
- -1 = O in location