Problem PA-1 (1 part)

Parameter Passing & Activation Frames

Part A: The following C function includes a call by value parameter and two call by reference parameters. Implement both the caller and called function in MIPS assembly. Use proper procedure linkage conventions.

```c
/* This program tests different parameter passing conventions. */
#include <stdio.h>

int main() {
    int X = 33;
    int Y = 55;
    int Z = 44;
    int P;
    int Conditional_Exchange(int *, int *, int);

    P = 0;
    Conditional_Exchange(&X, &Y, P);
    P = 1;
    Conditional_Exchange(&Y, &Z, P);
    printf("X = %d, Y = %d, Z = %d\n", X, Y, Z);
}

int Conditional_Exchange(int *A, int *B, int C) {
    int Temp;
    if (C) {
        Temp = *A;
        *A = *B;
        *B = Temp;
    }
    return(C);
}
```

Start with the shell MIPS code provided below.

```
# This program tests different parameter passing conventions.
.text
FP2:     add $30, $29, $00  # set caller's FP
addi $29, $29, -16  # create storage for locals X, Y, Z, P
addi $01, $00, 33    # load value 33
sw $01, -4($30)     # initialize x = 33
addi $01, $00, 55    # load value 55
sw $01, -8($30)     # initialize Y = 55
addi $01, $00, 44    # load value 44
sw $01, -12($30)    # initialize Z = 55

# set P = 0 and prepare to call CXchange here.
jal CXchange         # call CXchange(&X, &Y, P);

# set P = 1 and prepare to call CXchange again.
jal CXchange         # call CXchange(&Y, &Z, P);
add $29, $30, $00    # pop locals from stack
jr $31               # return to operating system
```

Note: the solution is several lines long and is best written and tested using MISASIM.