1. (40%) Define a C struct of your choice. Let us call it fun_struct. Then, write a C function to compare the contents of two variables of fun_struct. The function returns a 1 if the two variables are equal and a 0 otherwise. Write a simple main() to define and initialize the two fun_struct variables. Now, compile the program and generate its assembly code. Based on the assembly code, (a) please draw the layout of the activation record of your function, and (b) describe how the two fun_struct parameters and the elements in the fun_struct are referenced.

2. (25%) For the above program, if the C function takes two fun_struct variables as parameters, defines a local fun_struct variable and returns a fun_struct variable, observe how the assembly code generated by your compiler differs from that in the above.

3. (35%) Write a simple C function that contains a block, i.e., a segment of stations enclosed by { and } that declare some local variables. Let some local variables declared inside the block have the same name as those declared outside. Compile your program and generate the assembly code. Observe how your compiler implements the block and refers to its local variables.