Study Questions (Chapter 04 – Part 2)

1. The call-by-value program in part 2 of lecture 4 utilizes an FD stack. Modify the program to make it utilize an ED stack.
   Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
   Type, assemble, and run your modified program to make sure that it works properly.

2. The call-by-value program in part 2 of lecture 4 utilizes an FD stack. Modify the program to make it utilize an FA stack.
   Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
   Type, assemble, and run your modified program to make sure that it works properly.

3. The call-by-value program in part 2 of lecture 4 utilizes an FD stack. Modify the program to make it utilize an EA stack.
   Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
   Type, assemble, and run your modified program to make sure that it works properly.

4. The call-by-reference program in part 2 of lecture 4 utilizes an FD stack. Modify the program to make it utilize an ED stack.
   Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
   Type, assemble, and run your modified program to make sure that it works properly.

5. The call-by- reference program in part 2 of lecture 4 utilizes an FD stack. Modify the program to make it utilize an FA stack.
   Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
   Type, assemble, and run your modified program to make sure that it works properly.

6. The call-by- reference program in part 2 of lecture 4 utilizes an FD stack. Modify the program to make it utilize an EA stack.
   Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
   Type, assemble, and run your modified program to make sure that it works properly.