Problem 1

The caller, in this case alice(), pushes values of EAX, ECX, and EDX onto the stack. Alice() then pushes the argument onto the stack in reverse order, if args are a, b, and c, it pushes c, then b, and then a. alice() then calls bob(), pushing the return address onto the stack and then jumps to bob(). Bob() pushes the base pointer(EBP) and sets it to the current top of the stack, then pushes each local variable to the stack. In order to communicate the return value to alice(), bob() deallocates local variables, then restores ESI, EDI, EBX, and EBP by popping the saved values off the stack. It then pops the return address from the stack by calling ret, and then jumps back to the caller.

Problem2

1. Pushes EBP onto the stack
2. Copies stack pointer to ebp
3. Makes space on stack for local data
4. Put value 5 as second arg.
5. Put value 2 as first arg.
6. Call bob.
7. Bob pushes ebp onto stack
8. Copies stack pointer to ebp
9. Makes space on stack for local variables
10. Puts value 3 in z, the address of z is here now.
11. Puts the value of the second arg into eax
12. Adds eax to z, or in this case arg y to local variable z
13. Move this value to the return value
14. Return to main
15. Take return value, and store it into the address of a.