Eyup yayan, Kedir keyser og Siyar yuce

Task 1

1. The lines 5–13 in the code likely contain #define directives or constant declarations that establish meaningful names for various system call codes or other important values. By using names like SYS_EXIT, the code becomes more self-documenting, making it clearer that the number 1 refers to the system call for exiting a program.

2. Line 17 seems to be part of an assembly language program, and the resb directive is used to reserve bytes of memory. The command resb 4 reserves 4 bytes of memory for a variable named siffer, which can later be used for storing data like an integer.
3. The actual output to the console is handled by the int 80h instruction, which is a system interrupt in assembly language for Linux system calls. Prior lines prepare the system call by setting up the appropriate registers with the address of the message and the system call number for writing to the standard output (STDOUT).
4. After executing the instruction on line 43, the program counter would typically jump to the label or function that has been called. If the line is a call instruction, it would jump to the corresponding function's entry point.
5. The lessiffer block likely includes more than just lines 89 and 90. It probably spans from the label marking the start of the block to the next label or the ret instruction that ends the function.
6. The ret instruction on line 122 would return execution to the instruction immediately following the call to the Feil block, which is located somewhere in the _start section or main body of the code that initiated the call to lessiffer.