Task 1

1.

Lines 5-13 are constants for use in the rest of the code. Line 5 and 15 are just comments as we can tell by the leading ";". Their purpose is to tell us what the following lines of code are. Lines 6 and 7 assign values to two variables named "cr" and "lf". The values given to the variables are the values used for a line shift and a carrier return. These are used later in the code when we want to code a line shift or carrier return instead of directly using the values 13 and 10. Lines 8-13 also assign values to variables. Here we store values for various system calls in Linux systems that we need later in the code. We assign these so we can more easily perform the wanted system calls without having to remember the values.

2.

Line 17 reserves memory in the ".bss" data section of the program. "resb" means "reserve byte" and is a NASM directory that reserves an unutilized memory portion for the program. The full line means that we label the reserved memory "siffer" and we want the reserved memory to be 4 bytes.

3.

The lines that print out the message are lines 35-39. In these lines we set in the edx and ecx registers the length of the message and the address of our message to be printed. After that we load into registers ebx and and eax that we want our message to be sent to the standard output channel and for the system to write the message. Then when all our values are set in the appropriate registers we call the kernel to perform the printing of the message for us.

4.

line 43 executes the pogram counter jumps to line 86 where the start of lessiffer resides.

5.

Lines 86-122 is included in the lessiffer block.

6.

The ret instruction at line 122 is in the “feil” block that handles error in user input. When the line is executed the program counter jumps to the next line from where lessiffer was called from since its called from multiple places. For example if lessiffer is called from line 43 and then ret at line 122 is executed, the program counter will jump to line 44 and continue from there.