

Task 1

1. Explain Lines 5–13 and what is the purpose of these lines?

The EQU directive is used to defining constants in your code. In this case these constants are referring to system call numbers in NASM (32-bit system call numbers) and ASCII values. One could have written the system call numbers as numeric values (0, 1, 2 etc) while coding, but setting them as constants with names makes the code more readable and maintainable.

cr equ 13 → 13 is the ASCII value for the carriage return character

lf equ 10 → 10 is the ASCII value for the line feed character.

In the code these constants are used to generate new lines and move the cursor to a specific position.

SYS_EXIT equ 1 → In the code system call number 1 is used to exit the program. Usually the system call is placed in eax register.

SYS_READ equ 3 → In the code it is used to prepare the system for a read operation. Usually the system call is placed in eax register.

SYS_WRITE equ 4 → In the code the write system call is used to write data to the stdout stream and print it onto the console. Usually the system call is placed in eax register.

STDIN equ 0 → In the code its used to prepare the system to read from the console. "Standard input".

STDOUT equ 1 → In the code its used to prepare the register (usually ebx), and indicate that it will be used for output operations. "Standard output".

STDERR equ 2 → In the code this is used for "Standard error". Preparing the register (usually ebx) to be used for error operations.

These are system calls, and needs to be invoked by int 80h (to switch to kernel mode and interrupting the system).

2. What does Line 17 do? In particular, explain resb in this line.

siffer resb 4 is reserving 4 bytes of uninitialized data. The allocated storage space is named *siffer*.

RESB stands for "Reserves 1 byte". One should try to use the size that best fits the program. In the code resb 4 can hold one 32-bits integer.

3. Which lines in the program are supposed to print the messages below? Explain your answer.

```
Skriv to ensifrede tall skilt med mellomrom.  
Summen av tallene maa vaere mindre enn 10.
```

Line 35-38 is preparing the system to print the message, line 39 invokes the system calls. The message is printed after line 39.

mov edx , meldlen	← "meldlen" gives the length of "meld". This line puts the length (90) in register edx.
mov ecx , meld	← Pointer to the message to write.
mov ebx , STDOUT	← Preparing the system for output stream
mov eax , SYS_WRITE	← Preparing the system to write to the console / Printing the messages
int 80h	← Invokes the above system calls → After this interrupt "meld" is printed to terminal

4. After executing the instruction on Line 43, which line the program counter will jump to?

call lessiffer jumps to the function "lessiffer" on line 86.

5. Which lines are included in the lessiffer block?

The lines from 86-122 are included.

6. The block Feil on Line 113 is called conditionally from Lines 104 & 106. Where does the ret instruction on Line 122 return to?

It returns back to line 44, the code right after "call lessiffer". First the code test if the reading was OK. And since it is not, it sends you to "Slutt" and exits the program.