1 Comparision and jump instructions

Consider the following program:

```
section .data

section .text

        global _start

        _start:
                MOV eax, 3
                MOV ebx, 2

                CMP eax, ebx
                JL lesser
                JMP end

        end:
                MOV eax, 1
                INT 80h

        lesser:
                MOV eax, 1
                INT 80h
```

The **CMP** instruction basically performs subtraction between **eax** and **ebx**. If $\mathbf{eax} - \mathbf{ebx}$ results into a positive number then it will conclude that $\mathbf{eax} > \mathbf{ebx}$, if the result is negative then $\mathbf{eax} < \mathbf{ebx}$ and if the result is 0 then $\mathbf{eax} = \mathbf{ebx}$.

JL performs a signed comparision jump after a **CMP** if the destination operand(**eax**) is less than the source operand(**ebx**). Now, if **eax** is not less than **ebx** then the program will execute normally. The control will execute **JMP** instruction which performs an **unconditional** jump to the label **end**.

Labels are just user-defined names that are provided to make the code more readable, they are not functions. Meaning that if we have a label other than the **_start** label and we have not used any jump statements whatsoever, then the code inside of those labels will get executed **normally**.

1.1 More jump instructions

```
JE lesser
```

Jump to the label **lesser** if **eax** $=$ **ebx**.

```
JNE lesser
```

Jump to label **lesser** if **eax** $\neq$ **ebx**.

```
JG lesser
```

Jump to label **lesser** if **eax** $>$ **ebx**.

```
JGE lesser
```

Jump to label **lesser** if **eax** $\geq$ **ebx**.

```
JLE lesser
```

Jump to label **lesser** if **eax** $\leq$ **ebx**.

```
JZ lesser
```

Visit this link for this instruction.

```
JNZ lesser
```

Visit this link for this instruction.