EE309 Assignment 2

Raghav Singhal

19D070049

1 Question 1

```
lea ax, table[si]
```

Places the effective address obtained by adding the offset to the constant in AX.

```
mov ax, table[si]
```

Places the value present at the effective address obtained by adding the offset to the constant in AX. Let a value 37H be stored at effective address 0500.

```
lea ax, [500]
```

This command will place the effective address 0500 in AX.

```
mov ax, [500]
```

This command will place the value at the effective address, 37H in AX.

2 Question 2

- **JL/JNGE**
  - Flag Condition: $SF \neq OF$
  - Jump if operand 1 is lesser (not greater) than operand 2
  - If operand 1 is lesser than operand 2, and the subtraction overflows beyond -128, OF is 1 and SF is 0, thereby satisfying the condition.
  - If operand 1 is lesser than operand 2, and the subtraction does not overflows beyond -128, OF is 0 and SF is 1, thereby satisfying the condition.
- **JNL/JGE**
  - Flag Condition: $SF = OF$
  - Jump if operand 1 is not lesser (greater or equal) than operand 2

  - If operand 1 is not lesser than operand 2, and the subtraction overflows beyond +127, OF is 1 and SF is 1, thereby satisfying the condition.
  - If operand 1 is not lesser than operand 2, and the subtraction does not overflows beyond +127, OF is 0 and SF is 0, thereby satisfying the condition.

- **JLE/JNG**
  - Flag Condition: $(SF \neq OF)$ OR $(ZF = 1)$
  - Jump if operand 1 is lesser or equal (not greater) than operand 2
  - The lesser than condition has been discussed above; if it is true, the first term will be set.
  - If the 2 operands are equal, ZF will be set.
  - Thus, if either of the 2 terms is set, the condition is satisfied.

- **JNLE/JG**
  - Flag Condition: $(SF = OF)$ AND $(ZF = 0)$
  - Jump if operand 1 is neither less nor equal (greater) than operand 2
  - Operand 2 is greater or equal if $(SF = OF)$, ie. if the first term is set, as discussed earlier.
  - Since it must not be equal, ZF must not be set.
  - Thus, both conditions must be satisfied to satisfy the overall condition.

3 Question 3

The last 2 digits of my roll number are 4 and 9 respectively.
Properly commented code has been attached.

- The results obtained were as follows:
  - **Subtraction by 79**: 7, 0 at successive bytes
  - **Multiplication of obtained digits**: 0, 0 at successive bytes
  - **Dividing by 7**: Override the previous bytes to store 0, 0

- The flags obtainedwere as follows:
  - CF = 0, ZF = 1, SF = 0
  - OF = 0, PF = 1, AF = 0

- The processes used were as follows:

- **Subtraction**: **SUB** for subtraction followed by **DAS** for decimal adjust. The result was ANDed with 0FH for the first digit, and ANDed with 0F0H followed by shifting by 4 bits to the right for the second digit.
- **Multiplication**: **MUL** is used, followed by **AAM** for decimal adjust.
- **Division**: **DIV** is used. Since we need the digits at separate bytes, nothing else is required.

4 Question 4

```
;TABLE refers to starting address of the functions
MOV [52H], AH   ;save value of AH in at 52H
MOV AH, 00      ;move 0 to AH
MOV SI, AX      ;copy value of AX (AL since AH 0) to SI
CALL TABLE[SI]  ;call required function
```