Arithmetic Operations

- ADD

```
mov rax,0x1
add rax,0x1
```

- ADC
  - **when register is full adding 1 will be carried of and flag CF will be saved**
  - if carry flag is set then it will add with carry flag
  - `adc rax,0x1` rax will have 2 if CF is set other wise 1.

- SUB

```
mov rax,0x9
sub rax,0x4
```

- SBB
  - if carry flag is set then it will subtract addition 1 (carry flag) Other wise normal substract
  - `sbb rax,0x3` will substract 4 if CF is set Otherwise will substract 3

- INC
  - will increment by 1
  - `inc rax`

- DEC
  - will decrement by 1.
  - `dec rax`

- `clc` will clear any carry flag.
- `stc` will set the carry flag.
- `cmc` will coplimenting it that is (! reverse)

Code :

```
global _start

section .text
_start:

```

```
; register based addition

mov rax, 0x01
add rax, 0x01

mov rax, 0xffffffffffffffff
add rax, 0x01

mov rax, 0x09
sub rax, 0x04


; memory based addition

mov rax, qword [var1]
add qword [var4], rax

add qword [var4], 0x02


;  set / clear / complement carry flag

clc
stc
cmc

; add with carry

mov rax, 0
stc
adc rax, 0x1
stc
adc rax, 0x2

; subtract  with borrow

mov rax, 0x10
sub rax, 0x5
stc
sbb rax, 0x4

; increment and decrement
```

```
    inc rax
    dec rax

    ; exit the program gracefully

    mov rax, 0x3c
    mov rdi, 0
    syscall

section .data

    var1        dq      0x1
    var2        dq      0x1122334455667788
    var3        dq      0xffffffffffffffff
    var4    dq      0x0
```