- **PAF – Karachi Institute of Economics and Technology**

Course: Computer Organization & Assembly Language
Faculty: Yumna Shahzad

ASSIGNMENT#2

NIMRA JAMIL 10187

1-Write a coding to Jump to a label if 1st digit of your roll number is even.

answer :

```
.data
firstdigit word "1",0
.code
main proc
mov ax,firstdigit
and ax,1          ; low bit set?
jz EvenValue    ; jump if Zero flag set
```

2-Write a coding to Jump to a label if the 2nd digit of your roll number is not zero.

```
.data
secdigit byte "0",0
.code
main proc
mov al,secdigit
or al,al
jnz IsNotZero ; jump if not zero
```

5- Using only mov,add,sub,inc and dec,translate the following high level language assignment statements into assembly language where A,B and C are word variables.

a)A= -(A+1)

annswer :

```
mov ax,a
inc ax
neg ax
```

b)B= 3*B+7

answer :

```
mov ax,b
add ax,b
```

add ax,b
add ax,7
mov b,ax

6-Define the methodology used in stack frame?and how the stack frame is different from run timE stack, elaborate?

ANSWER :

Each capacity has nearby memory related with it to hold approaching boundaries, neighborhood factors, and (at times) brief factors. This district of memory is known as a stack outline and is apportioned on the cycle's stack. A casing pointer (the ebp register on intel x86 designs, rbp on 64-cycle structures) contains the base location of the capacity's casing. The code to get to nearby factors inside a capacity is created regarding balances to the edge pointer. The stack pointer (the esp register on intel x86 structures or rsp on 64-cycle designs) may change during the execution of a capacity as qualities are pushed or flown off the stack, (for example, pushing boundaries in readiness to calling another capacity). The casing pointer doesn't change all through the capacity.

This is what occurs during capacity (there may be slight contrasts among dialects/designs)

- Push the current estimation of the edge pointer (ebp/rbp). This spares it so we can reestablish it later.

- Move the current stack pointer to the casing pointer. This characterizes the beginning of the edge.

- Subtract the space required for the capacity's information from the stack pointer. Recollect that stacks develop from high memory to low memory. This gets the stack pointer past the space that will be utilized by the capacity so anything pushed onto the stack presently won't overwrite helpful qualities.

- Now execute the code for the capacity. References to nearby factors will be negative balances to the casing pointer .

- On exit from the capacity, duplicate the incentive from the edge pointer to the stack pointer (this clears up the space dispensed to the stack outline for the capacity) and pop the old edge pointer. This is refined by the "leave" guidance.

- Return from the technique through a "ret" guidance. This pops the return an incentive from the stack and moves execution to that address.

7-Find the factorial of user input number through recursion.

ANSWER :

```
.code
 main PROC

push 8
```

```
push eip
mov eip,ebp ;pointer registers
mov eax,[eip+8]
cmp eax,0
Ja loop1
mov eax,1
Jmp L2

loop1: dec  eax
push eax
;Recursive call returns.
Recursivecall:
mov ebx,[eip+8]
mul ebx
call writeDec
call crlf
exit
main ENDP
END main
```

☺ Good Luck ☺