**Name: Md. Rifat Ahmed**

**ID: 1931725042**

**Answer to the Question No: 1**

A Microprocessor is a CPU on a single chip that does not have any other things like memory, I/O units attached to it. On the other hand, a Microcontroller is made of a microprocessor, I/O peripheral units, ALU, registers and memory.

There are various models in Assembly language that differ in code size and memory size like tiny, small, compact, huge or large. But we normally use Small model which can use less than 64K for both code size and memory size.

If there are multiple procedures the Assembler can identify the driving procedures by checking which procedure is calling other procedures. For example, we normally use main procedure as the driving procedure and call other procedures through it.

**Answer to the Question No: 2**

.model small

.stack 200h

.data

array db 3, 4, 5, 6, 8

.code

main proc

mov ax, @data

mov ds, ax

lea si, array

mov cx,

mov ax, [si]

mov bx, 2

div bx

mov cx, 0 ;using to count Even

mov di, 0 ;using to count Odd

cmp dx, ax

jg Even

Even:

inc cx

Odd:

inc di

mov ah, 4ch

int 21h

main endp

end main

**Answer to the Question No: 3**

.model small

.stack 200h

.data

str1 db 'Rifat'

len1 equ $-str1

str2 db 'Ahmed'

len2 equ $-str2

str3 db len1+len2 dup(?)

len3 equ $-str3

.code

incre proc

inc dx

ret

incre endp

main proc

mov ax, @data

mov ds, ax

lea si, str1

lea di, str3

mov cx, 0

Traverse1:

cmp cx, len1

jz NewLoop

mov ax, [si]

mov [di], ax

inc cx

inc si

inc di

jmp Traverse1

NewLoop:

lea si, str2

mov cx, 0

Traverse2:

cmp cx, len2

jz SetupCount

mov ax, [si]

mov [di], ax

inc cx

inc si

inc di

jmp Traverse2

SetupCount:

mov dx, 0 ;using DX for Counting

mov cx, 0

lea si, str3

lea di, str3

mov bx, [di]

Count:

mov ax, [si]

inc si

cmp ax,bx

jz CountIncrement

CountIncrement:

call incre

cmp cx, len3

jz Terminate

inc cx

jmp Count

Terminate:

mov ah, 4ch

int 21h

main endp

end main

**Answer to the Question No: 4**

.model small

.stack 200h

.data

a equ 2h

b equ 3h

.code

multiplication proc

mul bx;

ret

multiplication endp

main proc

mov ax, @data

mov ds, ax

mov ax, a

mov bx, b

call multiplication

mov ah, 4ch

int 21h

main endp

end main