**Question#1:**

**Code:**

[org 0x0100] ; Program starts at offset 0x0100

jmp start ; Jump to start of program

studentID db 2, 2, 0, 4, 0, 7, 3, 2, 7 ; Student ID digits (excluding alphabets)

sum db 0 ; Memory location to store the sum of digits

res db 0 ; Result to store 0 (odd) or 1 (even)

start:

xor ax, ax ; Clear AX register (used for summing)

lea si, [studentID] ; Load address of the studentID array into SI

sum\_digits:

mov al, [si] ; Load the current digit from the array into AL

add [sum], al ; Add the current digit to the sum (stored in [sum])

inc si ; Move to the next digit in the array

cmp si, studentID + 9 ; Check if we have processed all 9 digits

jb sum\_digits ; If not, repeat the loop

mov al, [sum] ; Load the total sum from memory into AL

test al, 1 ; Test if the least significant bit (LSB) is 1

jz even ; If zero, the sum is even, jump to even label

odd:

mov dx, 0 ; If odd, store 0 in DX

jmp done ; Jump to done

even:

mov dx, 1 ; If even, store 1 in DX

done:

mov ah, 4Ch ; DOS interrupt to terminate program

int 21h ; Call DOS interrupt

**ScreenShort:**

**My VU ID is “bc230407844”.  
The sum of all numeric values is: 2 + 3 + 0 + 4 + 0 + 7 + 8 + 4 + 4 = 32.  
Now we convert the decimal 32 to hexadecimal, which is 20.  
As the sum is even, DX shows 0001 in the result.**

