**AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY**

****

**Department of Computer Science and Engineering**

Program: BSc in Computer Science and Engineering

Course Code: CSE 2214

Assignment No: 05

Date of Submission: 25.08.2025

Submitted by,

Name: Danial Hossain Dani

Student ID: 20230104058

Lab Section: B1

**1. Write a program that lets the user enter time in seconds and**

**outputs the time as hours, minutes, and seconds.**

**Answer:**

**.model small**

**.stack 100h**

**.data**

**msg1 db 'Enter time in seconds: $'**

**msg2 db 0dh,0ah,'Time (hh:mm:ss) = $'**

**input db 6,?,6 dup(?)**

**number dw ?**

**hour dw ?**

**minute dw ?**

**second dw ?**

**.code**

**main proc**

**mov ax, @data**

**mov ds, ax**

**mov ah, 9**

**lea dx, msg1**

**int 21h**

**mov ah, 0Ah**

**lea dx, input**

**int 21h**

**mov cl, [input+1]**

**mov si, offset input+2**

**xor ax, ax**

**convert\_loop:**

**cmp cl, 0**

**je done\_convert**

**mov bl, [si]**

**sub bl, '0'**

**xor bh, bh**

**mov dx, ax**

**mov ax, 10**

**mul dx**

**add ax, bx**

**inc si**

**dec cl**

**jmp convert\_loop**

**done\_convert:**

**mov number, ax**

**mov ax, number**

**xor dx, dx**

**mov bx, 3600**

**div bx**

**mov hour, ax**

**mov number, dx**

**mov ax, number**

**xor dx, dx**

**mov bx, 60**

**div bx**

**mov minute, ax**

**mov second, dx**

**mov ah, 9**

**lea dx, msg2**

**int 21h**

**mov ax, hour**

**call print\_two\_digits**

**mov dl, ':'**

**mov ah, 2**

**int 21h**

**mov ax, minute**

**call print\_two\_digits**

**mov dl, ':'**

**mov ah, 2**

**int 21h**

**mov ax, second**

**call print\_two\_digits**

**mov ah, 4Ch**

**int 21h**

**main endp**

**print\_two\_digits proc**

**push ax**

**cmp ax, 10**

**jae no\_zero**

**mov dl, '0'**

**mov ah, 2**

**int 21h**

**no\_zero:**

**pop ax**

**call print\_num**

**ret**

**print\_two\_digits endp**

**print\_num proc**

**push ax**

**push bx**

**push cx**

**push dx**

**mov bx, 10**

**mov cx, 0**

**cmp ax, 0**

**jne convert\_digits**

**mov dl, '0'**

**mov ah, 2**

**int 21h**

**jmp print\_done**

**convert\_digits:**

**xor dx, dx**

**digit\_loop:**

**div bx**

**push dx**

**inc cx**

**cmp ax, 0**

**jne digit\_loop**

**print\_loop:**

**pop dx**

**add dl, '0'**

**mov ah, 2**

**int 21h**

**loop print\_loop**

**print\_done:**

**pop dx**

**pop cx**

**pop bx**

**pop ax**

**ret**

**print\_num endp**

**end main**

**2.** **Write a program to find the greatest common divisor (GCD) of**

**two integers.**

**Answer:**

**.model small**

**.stack 100h**

**.data**

**num1 dw 48**

**num2 dw 18**

**.code**

**main proc**

**mov ax, @data**

**mov ds, ax**

**mov ax, num1**

**mov bx, num2**

**gcd\_loop:**

**cmp bx, 0**

**je done**

**xor dx, dx**

**div bx**

**mov ax, bx**

**mov bx, dx**

**jmp gcd\_loop**

**done:**

**add al, 30h**

**mov dl, al**

**mov ah, 2**

**int 21h**

**mov ah, 4Ch**

**int 21h**

**main endp**

**end main0**

**3. Write a program that starts with an initially undefined byte**

**array of maximum size 100, and lets the user insert single**

**characters into the array in such a way that the array is always**

**sorted in ascending order. The program should print a question**

**mark, let the user enter a character, and display the.array With**

**the new character Inserted. Input ends when the user hits the**

**F.SC key. Duplicate characters should be ignored.**

**Answer:**

**.model small**

**.stack 100h**

**.data**

**max\_size equ 100**

**prompt db '? $'**

**sortedmsg db 0dh,0ah,'sorted array: $'**

**space db ' $'**

**array db max\_size dup(?)**

**count db 0**

**.code**

**main proc**

**mov ax, @data**

**mov ds, ax**

**read\_loop:**

**lea dx, prompt**

**mov ah, 09h**

**int 21h**

**mov ah, 01h**

**int 21h**

**cmp al, '.'**

**je print\_final**

**mov cl, [count]**

**xor ch, ch**

**xor si, si**

**chk\_dup:**

**cmp cl, 0**

**je not\_dup**

**mov dl, [array+si]**

**cmp dl, al**

**je skip\_insert**

**inc si**

**dec cl**

**jnz chk\_dup**

**not\_dup:**

**mov cl, [count]**

**xor ch, ch**

**xor si, si**

**find\_pos:**

**cmp si, cx**

**jae insert\_here**

**mov dl, [array+si]**

**cmp al, dl**

**jl insert\_here**

**inc si**

**jmp find\_pos**

**insert\_here:**

**mov cl, [count]**

**xor ch, ch**

**mov di, cx**

**dec di**

**shift\_loop:**

**cmp di, si**

**jl place\_char**

**mov dl, [array+di]**

**mov [array+di+1], dl**

**dec di**

**jmp shift\_loop**

**place\_char:**

**mov [array+si], al**

**inc byte ptr [count]**

**jmp read\_loop**

**skip\_insert:**

**jmp read\_loop**

**print\_final:**

**lea dx, sortedmsg**

**mov ah, 09h**

**int 21h**

**mov cl, [count]**

**xor ch, ch**

**xor si, si**

**pr\_loop:**

**cmp si, cx**

**jae done**

**mov dl, [array+si]**

**mov ah, 02h**

**int 21h**

**lea dx, space**

**mov ah, 09h**

**int 21h**

**inc si**

**jmp pr\_loop**

**done:**

**mov ah, 4ch**

**int 21h**

**main endp**

**end main**