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
```