

Name: Promod Chandra Das
Id: 231002005
Subject: Microprocessors and microcontrollers
Subject code: CSE 304

Problem 1. Print Two Characters in Reverse Alphabetic Order

```
.model small
.stack 100h
.data
msg_1 DB 'Enter first capital letter: $'
msg_2 DB 0DH, 0AH, 'Enter second capital letter: $'
msg_3 DB 0DH, 0AH, 'Characters in reverse order: $'
.code
main proc
    mov ax, @data
    mov ds, ax

    ; Prompt for first character
    mov ah, 9
    lea dx, msg_1
    int 21h

    ; Read first character
    mov ah, 1
    int 21h
    mov bl, al ; Store first char in BL

    ; Prompt for second character
    mov ah, 9
    lea dx, msg_2
    int 21h

    ; Read second character
    mov ah, 1
    int 21h
    mov bh, al ; Store second char in BH

    ; Compare characters
    cmp bl, bh
    jg print_chars ; If BL > BH, already in reverse order
```

```

; Swap characters
xchg bl, bh

print_chars:
    mov ah, 9
    lea dx, msg_3
    int 21h

; Print first (greater) character
    mov ah, 2
    mov dl, bl
    int 21h

; Print space
    mov dl, ' '
    int 21h

; Print second (smaller) character
    mov dl, bh
    int 21h

; Exit
    mov ah, 4Ch
    int 21h

main endp
end main

```

Problem 2. Check if a Number is Odd or Even

```

.model small
.stack 100h
.data
msg_1 DB 'Enter a number: $'
msg_even DB 0DH, 0AH, 'The number is Even.$'
msg_odd DB 0DH, 0AH, 'The number is Odd.$'
.code
main proc
    mov ax, @data
    mov ds, ax

; Prompt for input
    mov ah, 9

```

```

    lea dx, msg_1
    int 21h

; Read number from user
    mov ah, 1
    int 21h
    sub al, '0' ; Convert ASCII to number

; Check even or odd
    test al, 1
    jz even_number

odd_number:
    mov ah, 9
    lea dx, msg_odd
    int 21h
    jmp exit

even_number:
    mov ah, 9
    lea dx, msg_even
    int 21h

exit:
    mov ah, 4Ch
    int 21h

main endp
end main

```

Problem 3. Find the Largest Number Between Two Numbers

```

.model small
.stack 100h
.data
msg_1 DB 'Enter first number: $'
msg_2 DB 0DH, 0AH, 'Enter second number: $'
msg_largest DB 0DH, 0AH, 'Largest number: $'
.code
main proc
    mov ax, @data
    mov ds, ax

```

; Prompt for first number

mov ah, 9

lea dx, msg_1

int 21h

; Read first number

mov ah, 1

int 21h

sub al, '0' ; Convert ASCII to number

mov bl, al ; Store in BL

; Prompt for second number

mov ah, 9

lea dx, msg_2

int 21h

; Read second number

mov ah, 1

int 21h

sub al, '0' ; Convert ASCII to number

mov bh, al ; Store in BH

; Compare numbers

cmp bl, bh

ja print_bl

mov dl, bh

jmp print_largest

print_bl:

mov dl, bl

print_largest:

mov ah, 9

lea dx, msg_largest

int 21h

mov ah, 2

add dl, '0' ; Convert to ASCII

int 21h

; Exit

mov ah, 4Ch

int 21h

```
main endp  
end main
```