Experiment – 3

Name: Nageshwar Prasad yadav
Reg. No.: AP21110011195
Section: CSE E

1. Write an assembly language program to find whether two 16-bit numbers are equal or not and print the appropriate message.

```
data SEGMENT
    n1 DW 1234h
    n2 DW 0A23h
    mes1 DB "Both the numbers are same $"
    mes2 DB "Both the numbers are different $"
data ENDS

code SEGMENT
    ASSUME cs:code, ds:data
Start: MOV AX, data
       MOV DS, AX
       MOV AX, n1
       CMP AX, n2
       MOV AH, 9
       JNZ noteq
       LEA DX, mes1
       JMP over
noteq: LEA DX, mes2
over: INT 21h
      MOV AX, 4C00h
```

```
    INT 21h
code ENDS
    END Start
```

2. Write Assembly language program to find the minimum of two 16-bit numbers.

```
ORG 100h
MOV AH, 9
LEA DX, prompt1
INT 21h
MOV AH, 1
INT 21h
SUB AL, 30h
MOV BX, AX

MOV AH, 9
LEA DX, prompt2
INT 21h

MOV AH, 1
```

```
INT 21h
SUB AL, 30h
MOV AX, BX
CMP AX, BX
JMP skip_swap
XCHG AX, BX
skip_swap: MOV CX, AX

MOV AH, 9
LEA DX, result
INT 21h
MOV AH, 2
INT 21h
MOV DL, 0Dh
INT 21h
MOV DL, 0Ah
INT 21h

MOV AH, 4Ch
INT 21h
prompt1 db 'Enter the first number: $'
prompt2 db 'Enter the second number: $'
result db 'The minimum value is: $'
```

3. Write an assembly language program to check whether the given 16-bit number is even or odd and print the appropriate message.

```
ORG 100h
ORG 100h
```

```
MOV AH, 9
LEA DX, prompt
INT 21h

MOV AH, 1
INT 21h
MOV BX, AX

AND BX, 1
JZ even
JMP odd

even:
MOV AH, 9
LEA DX, even_msg
JMP print_msg

odd:
MOV AH, 9
LEA DX, odd_msg

print_msg:
INT 21h

MOV AH, 4Ch
INT 21h

prompt db 'Enter a 16-bit number: $'
even_msg db 'The number is even.$'
```

```
odd_msg db 'The number is odd.$'

END
```

4. Write a program to find the sum of first N positive integers.

```
ORG 100h

MOV AH, 9
LEA DX, prompt
INT 21h

MOV AH, 1
INT 21h
SUB AL, 30h
MOV BL, AL

CMP BL, 0
JBE invalid_input

MOV CX, 0
MOV AX, 0
sum_loop:
    ADD AX, CX
    INC CX
    CMP CX, BX
    JLE sum_loop
```

```
MOV AH, 9
LEA DX, result
INT 21h
MOV AH, 2
MOV DL, ' '
INT 21h
MOV AH, 2
INT 21h
MOV DL, 0Dh
INT 21h
MOV DL, 0Ah
INT 21h

MOV AH, 4Ch
INT 21h

invalid_input:
MOV AH, 9
LEA DX, error
INT 21h
JMP exit_program

prompt db 'Enter a positive integer: $'
error db 'Error: Invalid input.$'
result db 'The sum of the first %d positive integers is %d.$'

exit_program:
RET
```