**Name: MD. Shamim Mia**

**ID: 19101532**

**Section: 08**

**Assignment 3**

**Task 01:**

.MODEL SMALL

.STACK 100h

.DATA

msg db "All the given numbers are even$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

;CH will contain the sum of odd numbers and CL will contain the count of odd numbers

MOV CL, 0

MOV CH, 0

;taking the first digit as input and moving the input to BH

MOV AH, 1

INT 21h

MOV BH, AL

;performing division to find out the remainder

MOV AH, 0

MOV BL, 2

DIV BL

;checking whether the number is odd or even

CMP AH, 0

JE TAKE\_INPUT2

ADD CH, BH

INC CL

TAKE\_INPUT2:

;going to new line where cursor will be at the beginning

MOV AH, 2

MOV DL, 0Dh

INT 21h

MOV DL, 0Ah

INT 21h

;taking the second digit as input and moving the input to BH

MOV AH, 1

INT 21h

MOV BH, AL

;performing division to find out the remainder

MOV AH, 0

MOV BL, 2

DIV BL

;checking whether the number is odd or even

CMP AH, 0

JE TAKE\_INPUT3

ADD CH, BH

INC CL

TAKE\_INPUT3:

;going to new line where cursor will be at the beginning

MOV AH, 2

MOV DL, 0Dh

INT 21h

MOV DL, 0Ah

INT 21h

;taking the third digit as input and moving the input to BH

MOV AH, 1

INT 21h

MOV BH, AL

;performing division to find out the remainder

MOV AH, 0

MOV BL, 2

DIV BL

;checking whether the number is odd or even

CMP AH, 0

JE PRINT\_OUTPUT

ADD CH, BH

INC CL

PRINT\_OUTPUT:

; going to new line where cursor will be at the beginning

MOV AH, 2

MOV DL, 0Dh

INT 21h

MOV DL, 0Ah

INT 21h

;checking if there is any odd number or not

CMP CL, 0

JE NO\_ODD

;performing division to find out the mean

MOV AL, CH

MOV AH, 0

DIV CL

;printing the output

MOV DL, AL

MOV AH, 2

INT 21h

JMP EXIT

NO\_ODD:

LEA DX, msg

MOV AH, 9

INT 21h

EXIT:

MOV AX, 4C00h

INT 21h

MAIN ENDP

END MAIN

**Task 02:**

.MODEL SMALL

.STACK 100h

.DATA

msg1 db "Good Morning$"

msg2 db "Good Afternoon$"

msg3 db "Good Night$"

msg4 db "Invalid$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

;keeping the hour at AL and checking the conditions

MOV AL, 5

CMP AL, 0

JL LOAD\_MSG4

CMP AL, 12

JL LOAD\_MSG1

CMP AL, 20

JL LOAD\_MSG2

CMP AL, 23

JLE LOAD\_MSG3

LOAD\_MSG4:

LEA DX, msg4

JMP PRINT

LOAD\_MSG1:

LEA DX, msg1

JMP PRINT

LOAD\_MSG2:

LEA DX, msg2

JMP PRINT

LOAD\_MSG3:

LEA DX, msg3

PRINT:

MOV AH, 9

INT 21h

MOV AX, 4C00h

INT 21h

MAIN ENDP

END MAIN

**Task 03:**

.MODEL SMALL

.STACK 100h

.DATA

msg1 db "Leap year$"

msg2 db "Not a leap year$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

;storing the year in CX register

MOV CX, 2022

MOV AX, CX

MOV DX, 0

;dividing by 400 to check whether it's leap year or not.

MOV BX, 400

DIV BX

CMP DX, 0

JE LOAD\_MSG1

;again storing the values and dividing by 100. if it's divisible then not a leap year.

MOV AX, CX

MOV DX, 0

MOV BX, 100

DIV BX

CMP DX, 0

JE LOAD\_MSG2

;again storing the values and checking whether it's divisible by 4 or not

MOV AX, CX

MOV DX, 0

MOV BX, 4

DIV BX

CMP DX, 0

JE LOAD\_MSG1

;if it's not divisible then it's not a leap year.

LOAD\_MSG2:

LEA DX, msg2

JMP PRINT

LOAD\_MSG1:

LEA DX, msg1

PRINT:

MOV AH, 9

INT 21h

MOV AX, 4C00h

INT 21h

MAIN ENDP

END MAIN

**Task 04:**

.MODEL SMALL

.STACK 100h

.DATA

msg1 db "equilateral$"

msg2 db "isosceles$"

msg3 db "neither$"

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

;storing 3 sides in 3 register AX, BX, CX

MOV AX, 5

MOV BX, 9

MOV CX, 8

;checking if first 2 sides are equal or not

CMP AX, BX

JE CHECK\_EQUI

;first 2 sides are not equal so checking if any of them are equal to the third one or not

CMP AX, CX

JE LOAD\_MSG2

CMP BX, CX

JE LOAD\_MSG2

;all the sides are unequal

LOAD\_MSG3:

LEA DX, MSG3

JMP PRINT

;first two sides are equal. now checking if the third side is equal or not

CHECK\_EQUI:

CMP BX, CX

JE LOAD\_MSG1

LOAD\_MSG2:

LEA DX, msg2

JMP PRINT

LOAD\_MSG1:

LEA DX, msg1

PRINT:

MOV AH, 9

INT 21h

MOV AX, 4C00h

INT 21h

MAIN ENDP

END MAIN