**ADIT LUHADIA**

**190911112**

**IT A**

**COMP LAB WEEK 3**

1. GCD of four unsigned words using procedure

; GCD of four unsigned words using procedure

DATA SEGMENT

    NUM DW 4,9,5,50

    GCD DW 3 DUP(?)

DATA ENDS

CODE SEGMENT

    ASSUME CS:CODE,DS:DATA

    START:

        MOV DX,DATA

        MOV DS,DX

        LEA SI,GCD

        MOV AX,NUM ; AX=0004

        MOV BX,NUM+2 ; BX=0009

        CALL GCDPROC ; gcd of 9,4

        MOV AX,NUM+4 ; AX=0005

        MOV BX,NUM+6 ; BX=0050

        CALL GCDPROC ; gcd of 5,50

        ;FINAL GCD

        MOV AX,GCD

        MOV BX,GCD+2

        CALL GCDPROC

        MOV AH,4CH

        INT 21H

        GCDPROC:

            CMP AX,BX

            JZ EXIT

            JB BIGAX; if AX IS BELOW BX, WE WANT TO MAKE AX BIGGER

            ;AX=0090 BX=0040

            MOV DX,0

            DIV BX

            CMP DX,0

            JZ EXIT

            MOV AX,BX

            MOV BX,DX

            CALL GCDPROC

            JMP EXIT

            BIGAX:

                XCHG AX,BX

                JMP GCDPROC

            EXIT: MOV [SI],BX ; GCD FOUND

                  INC SI

                  INC SI

            RET

CODE ENDS

END START

Graphical user interface

Description automatically generated

2. LCM of four unsigned bytes using function which finds LCM of two unsigned bytes.

; LCM of four unsigned bytes using function which finds LCM of two unsigned bytes.

DATA SEGMENT

        NUM DB 25, 15, 21, 30 ; 4 unsigned numbers

        LCM1 DW ?

        LCM DW ?

DATA ENDS

STACK SEGMENT

        DW 100 DUP(?)

        TOS LABEL WORD

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACK

START:  MOV AX, DATA

        MOV DS, AX

        MOV AX, STACK

        MOV SS, AX

        LEA SP, TOS

        MOV AH, 0

        MOV AL, NUM

        MOV BL, NUM+1

        CALL LCMPROC

        MOV AX, LCM

        MOV LCM1, AX

        MOV AH, 0

        MOV AL, NUM+2

        MOV BH, 0

        MOV BL, NUM+3

        CALL LCMPROC

        MOV AX, LCM1

        MOV BX, LCM

        CALL LCMPROC

        MOV AH, 4CH

        INT 21H

LCMPROC PROC

BACK:   PUSH AX

        DIV BL

        CMP AH, 0

        JZ DOWN

        POP AX

        ADD AL, NUM

        ADC AH, 0

        JMP BACK

DOWN:   POP LCM

        RET

LCMPROC ENDP

CODE ENDS

END START

Graphical user interface

Description automatically generated with low confidence

3. Factorial of unsigned byte using recursion

; Factorial of unsigned byte using recursion

STACK SEGMENT

        STK DW 100 DUP(?)

        TOS LABEL WORD

STACK ENDS

DATA SEGMENT

        NUM DB 07

        RES DW ?

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACK

START:  MOV AX, DATA

        MOV DS, AX

        MOV AX, STACK

        MOV SS, AX

        LEA SP, TOS

        MOV AL, NUM

        MOV AH, 0

        CALL FACT

        MOV AH, 4CH

        INT 21H

FACT PROC

        CMP AX, 01H

        JE EXIT

        PUSH AX

        DEC AX

        CALL FACT

        POP AX

        MUL RES

        MOV RES, AX

        RET

EXIT:   MOV RES, 01

        RET

FACT ENDP

CODE ENDS

END START

Graphical user interface

Description automatically generated

4. Linear search in an array of 10 unsigned Words

; Linear search in an array of 10 unsigned Words

PRINTSTR MACRO MSG

        LEA DX, MSG

        MOV AH, 09H

        INT 21H

ENDM

DATA SEGMENT

        ARRAY DW 19H, 20, 21H, 22H, 10H, 11H, 25H, 62H, 12H,09H

        M2 DB 'FOUND $'

        M3 DB 'NOT FOUND $'

        ELE DW 45H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:  MOV AX, DATA

        MOV DS, AX

        MOV CX, 10

        MOV SI, 0

        MOV AX, ELE

BACK1:  CMP AX, ARRAY[SI]

        JZ DOWN

        INC SI

        INC SI

        LOOP BACK1

        PRINTSTR M3

        JMP EXIT1

DOWN:   PRINTSTR M2

EXIT1:  MOV AH, 4CH

        INT 21H

CODE ENDS

END START

Text

Description automatically generated

5. Sorting of Signed Words Using Bubble sort

; Sorting of Signed Words Using Bubble sort

DATA SEGMENT

        ARRAY DW 34H, 78H, 56H, 47H, 22H

DATA ENDS

STACK SEGMENT

        DW 100 DUP(?)

        TOS LABEL WORD

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:STACK

START:  MOV AX, DATA

        MOV DS, AX

        MOV AX, STACK

        MOV SS, AX

        LEA SP, TOS

        MOV CX, 4

BACK1:  PUSH CX

        LEA SI, ARRAY

BACK:   MOV AX, [SI]

        CMP AX, [SI+2]

        JC DOWN

        XCHG AX, [SI+2]

        MOV [SI], AX

DOWN:   INC SI

        INC SI

        LOOP BACK

        POP CX

        LOOP BACK1

        MOV AH, 4CH

        INT 21H

CODE ENDS

END START

Graphical user interface

Description automatically generated