Assignment 1

A1-1)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

.MODEL SMALL

.STACK 100H

.DATA

MS DB "ENTER A CHARACTER: $"

MT DB 0DH,0AH,"THE CHARACTER IS: $"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MS

MOV AH,1H

INT 21H

MOV BL,AL

PRINT MT

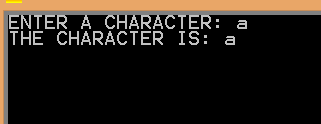
MOV AH,2H

MOV DL,BL

INT 21H

MAIN ENDP

END MAIN



A1-2)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

.MODEL SMALL

.STACK 100H

.DATA

MS DB 0DH,0AH,"ENTER AN UPPER CASE CHARACTER: $"

MT DB 0DH,0AH,"THE LOWER CASE CHARACTER IS: $"

WR DB 0DH,0AH,"WRONG INPUT$"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

LEV: PRINT MS

MOV AH,1H

INT 21H

CMP AL,41H

JL NO

CMP AL,5AH

JG NO

ADD AL,20H

MOV BL,AL

PRINT MT

MOV AH,2H

MOV DL,BL

INT 21H

JMP GO

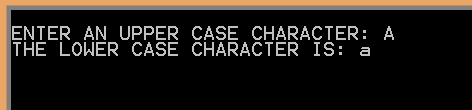
NO: PRINT WR

JMP LEV

GO:NOP

MAIN ENDP

END MAIN



A1-3)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

.MODEL SMALL

.STACK 100H

.DATA

N DW 1AH

SPC DB " $"

LN DB 0DH,0AH,"$"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV CX,N

MOV AX,41H

LEV:MOV DL,AL

MOV AH,2H

INT 21H

PUSH AX

PRINT SPC

POP AX

XOR AH,AH

INC AL

LOOP LEV

PRINT LN

MOV CX,N

MOV AX,61H

LEV1:MOV DL,AL

MOV AH,2H

INT 21H

PUSH AX

PRINT SPC

POP AX

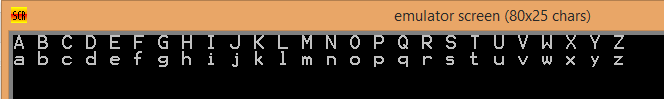
XOR AH,AH

INC AL

LOOP LEV1

MAIN ENDP

END MAIN



A1-4)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

.MODEL SMALL

.STACK 100H

.DATA

MS DB "MY NAME IS: Dibyayan Mondal $"

.CODE

MAIN PROC

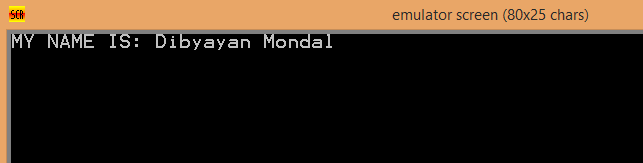
MOV AX,@DATA

MOV DS,AX

PRINT MS

MAIN ENDP

END MAIN



A1-5)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

A DW 1

B DW 1

N DW 0

MS DB 0DH,0AH,"ENTER HOW MANY FIBONACCI NUMBERS DO YOU WANT: $"

ANS DB 0DH,0AH,"THE REQUIRED FIBONACCI NUMBERS ARE$"

ARR DB " --> $"

CRS DB "X$"

LN DB 0DH,0AH,"$"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MS

READ N,U1,U2

MOV CX,N

PRINT ANS

PRINT LN

CMP CX,1H

JNE CHK

MOV AH,2H

MOV DL,31H

INT 21H

PRINT ARR

JMP GO

CHK: MOV AH,2H

MOV DL,31H

INT 21H

PRINT ARR

MOV AH,2H

MOV DL,31H

INT 21H

PRINT ARR

CMP CX,2H

JE GO

DEC CX

DEC CX

ADDI: MOV AX,A

ADD AX,B

MOV BX,B

MOV A,BX

MOV B,AX

PUSH CX

PRINTMUL AX,G1,G2

PRINT ARR

POP CX

LOOP ADDI

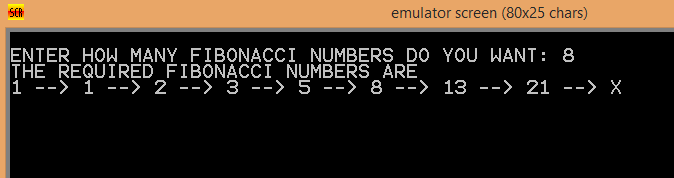
GO:PRINT CRS

MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN



A1-7)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,0AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,00H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

N DW 0

M DW 0

MS DB 0DH,0AH,"ENTER NUMBER: $"

PRI DB 0DH,0AH,"PRIME$"

NPRI DB 0DH,0AH,"NOT PRIME$"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MS

READ N,U1,U2

MOV AX,N

CMP AX,01H

JE PR

CMP AX,02H

JE PR

CMP AX,03H

JE PR

MOV BX,02H

DIV BX

CMP DX,00H

JE NPR

MOV BX,3H

LEV: MOV AX,N

XOR DX,DX

DIV BX

CMP DX,00H

JE NPR

INC BX

INC BX

CMP BX,N

JL LEV

PR: PRINT PRI

JMP GO

NPR: PRINT NPRI

GO: NOP

MAIN ENDP

END MAIN



A1-8)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

N DW ?

M DW ?

GCD DW ?

MS DB "ENTER THE 1ST NUMBER: $"

MT DB 0DH,0AH,"ENTER THE 2ND NUMBER: $"

LCM DB 0DH,0AH,"THE LCM OF THE 2 NOS IS: $"

HCF DB 0DH,0AH,"THE HCF OF THE 2 NOS IS: $"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MS

READ N,G1,G2

PRINT MT

READ M,B1,B2

PRINT HCF

MOV AX,N

MOV BX,M

CMP AX,BX

JG GO

SWAP: MOV CX,BX

MOV BX,AX

MOV AX,CX

GO: XOR DX,DX

XOR AH,AH

DIV BX

CMP DX,00H

JE HC

MOV AX,BX

MOV BX,DX

JMP GO

HC: MOV AX,BX

MOV GCD,AX

PRINTMUL AX,Y1,W2

PRINT LCM

MOV AX,N

MUL M

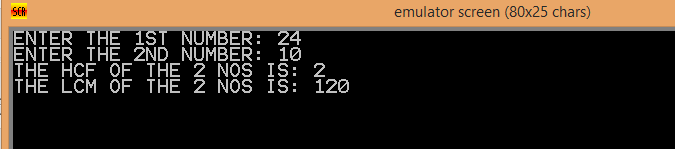
MOV BX,GCD

DIV BX

PRINTMUL AX,V3,B8

MAIN ENDP

END MAIN



A1-9)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

TEMP DW 0

MULT DW 1

MS DB "ENTER THE HEXA-DECIMAL NUMBER: $"

MT DB 0DH,0AH,"THE EQUIVALENT DECIMAL NUMBER IS: $"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MS

XOR CX,CX

LEV:MOV AH,1H

INT 21H

CMP AL,0DH

JE GO

XOR AH,AH

PUSH AX

INC CX

JMP LEV

GO:PRINT MT

OK: POP AX

SUB AX,30H

CMP AX,9H

JL NXT

SUB AX,7H

NXT:MUL MULT

MOV BX,TEMP

ADD AX,BX

MOV TEMP,AX ;TEMP=AX\*BX+TEMP

MOV AX,MULT

MOV DX,10H

MUL DX

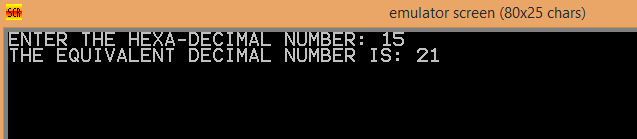
MOV MULT,AX ;MULT=MULT\*16

LOOP OK

PRINTMUL TEMP,B0,V2

MAIN ENDP

END MAIN



A1-10)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

N DW 0

MS DB "ENTER THE NUMBER: $"

FAC DB 0DH,0AH,"THE FACTORIAL IS: $"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MS

READ N,G1,G2

PRINT FAC

MOV CX,N

CMP CX,1H

JNE GO

MOV DL,31H

MOV AH,2H

INT 21H

JMP OK

GO:DEC CX

MOV BX,2

MOV AX,1

LEV: MUL BX

INC BX

LOOP LEV

PRINTMUL AX,U1,U2

OK: NOP

MAIN ENDP

END MAIN



ASSIGNMENT 2

A2-1)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

NUM DW 100 DUP(0)

N DW 0

M DW 0

MSG1 DB 10,13,'PLEASE ENTER THE NO OF ELEMENTS: $'

MSG2 DB 10,13,'ENTER THE ELEMENT: $'

MSG3 DB 10,13,'THE GIVEN ARRAY BEFORE SORTING: $'

MSG4 DB ' -> $'

MSG5 DB 10,13,'THE CURRENT STATE OF THE ARRAY: $'

MSG6 DB 'X$'

MSG7 DB 10,13,'THE FINAL ARRAY AFTER SORTING: $'

MSG8 DB 10,13,' SORTING IN ASCENDING ORDER$'

MSG9 DB 10,13,' SORTING IN DESCENDING ORDER$'

;PROGRAM TO IMPLEMENT BUBBLE SORT

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG1

READ N,JUMP1,JUMP2

MOV CX,N

MOV AX,N

DEC AX

MOV M,AX

MOV SI,0000H

LOOP1: PRINT MSG2

READ NUM[SI],JUMP3,JUMP4

ADD SI,02H

LOOP LOOP1

PRINT MSG3

CALL DISPLAY

PRINT MSG8

XOR CX,CX

OUTER: MOV SI,0000H

PUSH CX

XOR CX,CX

INNER: MOV DI,SI

MOV AX,NUM[SI]

ADD SI,02H

CMP AX,NUM[SI]

JL CHECK

MOV BX,NUM[SI]

MOV NUM[SI],AX

MOV NUM[DI],BX

CHECK: INC CX

CMP CX,M

JL INNER

PRINT MSG5

CALL DISPLAY

POP CX

INC CX

CMP CX,M

JL OUTER

PRINT MSG7

CALL DISPLAY

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

PRINT MSG9

XOR CX,CX

OUTER1: MOV SI,0000H

PUSH CX

XOR CX,CX

INNER1: MOV DI,SI

MOV AX,NUM[SI]

ADD SI,02H

CMP AX,NUM[SI]

JG CHECK1

MOV BX,NUM[SI]

MOV NUM[SI],AX

MOV NUM[DI],BX

CHECK1: INC CX

CMP CX,M

JL INNER1

PRINT MSG5

CALL DISPLAY

POP CX

INC CX

CMP CX,M

JL OUTER1

PRINT MSG7

CALL DISPLAY

MOV AH,4CH

INT 21H

MAIN ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL NUM[SI],L5,L6

PRINT MSG4

ADD SI,02H

POP CX

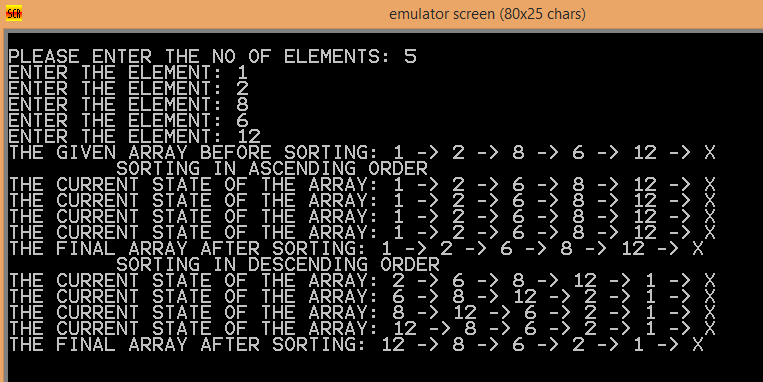
LOOP L4

PRINT MSG6

RET

DISPLAY ENDP

END MAIN



A2-2)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

NUM DW 100 DUP(0)

N DW 0

M DW 0

MSG1 DB 10,13,'PLEASE ENTER THE NO OF ELEMENTS: $'

MSG2 DB 10,13,'ENTER THE ELEMENT: $'

MSG3 DB 10,13,'THE GIVEN ARRAY BEFORE SORTING: $'

MSG4 DB ' -> $'

MSG5 DB 10,13,'THE CURRENT STATE OF THE ARRAY: $'

MSG6 DB 'X$'

MSG7 DB 10,13,'THE FINAL ARRAY AFTER SORTING: $'

;PROGRAM TO IMPLEMENT SELECTION SORT

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG1

READ N,JUMP1,JUMP2

MOV CX,N

MOV AX,N

DEC AX

MOV M,AX

MOV SI,0000H

LOOP1: PRINT MSG2

READ NUM[SI],JUMP3,JUMP4

ADD SI,02H

LOOP LOOP1

PRINT MSG3

CALL DISPLAY

MOV SI,0000H

XOR CX,CX

;DI is used for min\_index

;SI is used for comparing the array values, comparing a[SI] with a[SI+1]

;CX is used for both the loops. It's value is being saved in the stack

;for future use

OUTERLOOP: MOV AX,NUM[SI]

MOV DI,SI

PUSH CX

PUSH SI

MOV SI,DI

INNERLOOP:

ADD SI,02H

CMP AX,NUM[SI]

JL CHECK

MOV AX,NUM[SI]

MOV DI,SI

CHECK: INC CX

CMP CX,M

JL INNERLOOP

POP SI

POP CX

;THE SWAPPING PROCEDURE

MOV BX,NUM[SI]

MOV NUM[DI],BX ;a[min\_index]=a[i]

MOV NUM[SI],AX ;a[i]=a[min\_index]

;push CX and SI into stack so that their value can be got after display

PUSH CX

PUSH SI

;printing the current state

PRINT MSG5

CALL DISPLAY

;retrieving CX and SI from stack

POP SI

POP CX

ADD SI,02H ;i++

INC CX

CMP CX,M

JL OUTERLOOP

PRINT MSG7

CALL DISPLAY

MOV AH,4CH

INT 21H

MAIN ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL NUM[SI],L5,L6

PRINT MSG4

ADD SI,02H

POP CX

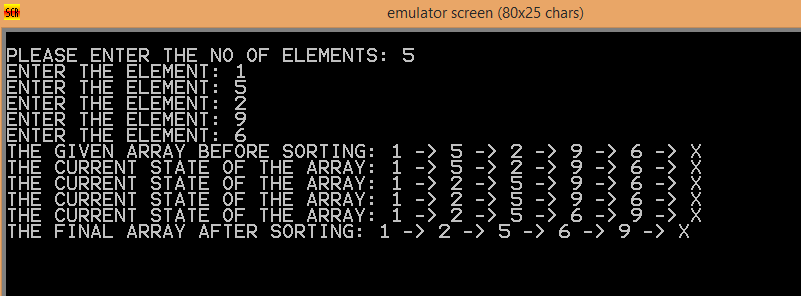
LOOP L4

PRINT MSG6

RET

DISPLAY ENDP

END



A2-3)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

NUM DW 100 DUP(0)

N DW 0

M DW 0

MSG1 DB 10,13,'PLEASE ENTER THE NO OF ELEMENTS: $'

MSG2 DB 10,13,'ENTER THE ELEMENT: $'

MSG3 DB 10,13,'THE GIVEN ARRAY BEFORE SORTING: $'

MSG4 DB ' -> $'

MSG5 DB 10,13,'THE CURRENT STATE OF THE ARRAY: $'

MSG6 DB 'X$'

MSG7 DB 10,13,'THE FINAL ARRAY AFTER SORTING: $'

MSG8 DB 10,13,' SORTING IN ASCENDING ORDER$'

MSG9 DB 10,13,

; PROGRAM TO IMPLEMENT INSERTION SORT

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG1

READ N,JUMP1,JUMP2

MOV CX,N

MOV AX,N

DEC AX

MOV M,AX

MOV SI,0000H

LOOP1: PRINT MSG2

READ NUM[SI],JUMP3,JUMP4

ADD SI,02H

LOOP LOOP1

PRINT MSG3

CALL DISPLAY

PRINT MSG8

MOV SI,0000H

ADD SI,02H

XOR CX,CX

INC CX

OUTER:

PUSH CX

MOV AX,NUM[SI]

MOV DI,SI

SUB DI,02H

PUSH SI

INNER: CMP AX,NUM[DI]

JG CM

MOV BX,NUM[DI];BX=A[j]

MOV NUM[SI],BX;A[i]=BX=A[j]

CHECK: MOV SI,DI;i=j

SUB DI,02H;j--

CMP DI,0000H

JGE INNER

CM: CMP AX,NUM[SI]

JG LAB

MOV NUM[SI],AX

LAB: PRINT MSG5

CALL DISPLAY

POP SI

ADD SI,02H

POP CX

INC CX

CMP CX,N

JL OUTER

PRINT MSG7

CALL DISPLAY

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

PRINT MSG9

MOV SI,0000H

ADD SI,02H

XOR CX,CX

INC CX

OUTER1:

PUSH CX

MOV AX,NUM[SI]

MOV DI,SI

SUB DI,02H

PUSH SI

INNER1: CMP AX,NUM[DI]

JL CM1

MOV BX,NUM[DI];BX=A[j]

MOV NUM[SI],BX;A[i]=BX=A[j]

CHECK1: MOV SI,DI;i=j

SUB DI,02H;j--

CMP DI,0000H

JGE INNER1

CM1: CMP AX,NUM[SI]

JG LAB1

MOV NUM[SI],AX

LAB1: PRINT MSG5

CALL DISPLAY

POP SI

ADD SI,02H

POP CX

INC CX

CMP CX,N

JL OUTER1

PRINT MSG7

CALL DISPLAY

MOV AH,4CH

INT 21H

MAIN ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL NUM[SI],L5,L6

PRINT MSG4

ADD SI,02H

POP CX

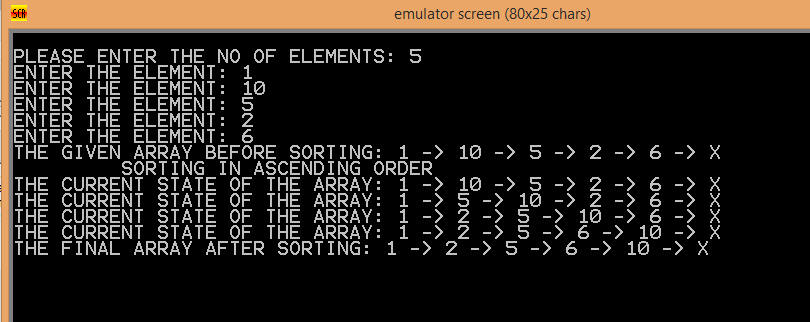
LOOP L4

PRINT MSG6

RET

DISPLAY ENDP

END



A2-4)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

NUM DW 100 DUP(0)

N DW 0

M DW 0

MAX DW 0

KEY DW 0

MSG1 DB 10,13,'ENTER THE ELEMENT: $'

MSG2 DB 10,13,'THE GIVEN ARRAY: $'

MSG3 DB 10,13,'THE ELEMENT IS FOUND IN THE POSITION(S): $'

MSG4 DB ' -> $'

MSG6 DB 'X$'

MSG8 DB 10,13,'SORRY, THE ELEMENT NOT FOUND!!!$'

MSG7 DB 10,13,'PLEASE ENTER THE ELEMENT TO BE SEARCHED: $'

MSG5 DB 10,13,'ENTER THE SIZE OF THE ARRAY: $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG5

READ N,JUMP1,JUMP2

MOV CX,N

MOV AX,N

DEC AX

MOV M,AX ;M = (n - 1)

ADD AX,AX

MOV MAX,AX

MOV SI,0000H

LOOP1: PRINT MSG1

READ NUM[SI],JUMP3,JUMP4

ADD SI,02H

LOOP LOOP1

PRINT MSG2

CALL DISPLAY

PRINT MSG7

READ KEY,JUMP5,JUMP6

PRINT MSG6

PRINT MSG3

MOV CX,KEY

MOV SI,0000H

MOV BX,00H

MOV DX,M

;bx will hold the value of l = 0 initially

;dx will hold the value of h = (n - 1) initially

;ax will hold mid value = (l + h) / 2

OUTER: CMP BX,DX ;if(l > h)

JA NOT\_FOUND ; jump to not possible

MOV AX,BX ;ax = l

ADD AX,DX ;ax = (h + l)

SHR AX,1 ;ax = (h + l) / 2

MOV SI,AX ;i = ax

ADD SI,SI

CMP CX,NUM[SI] ;cmp key, num[mid]

JL GREATER

JE FOUND

LOWER: ADD AX,01

MOV BX,AX ;l = (m + 1)

JMP OUTER

GREATER: SUB AX,01H

MOV DX,AX ;h = (m - 1)

JMP OUTER

FOUND: MOV BX,AX ;bx = m

;code to print the first mid position encountered

MOV DL,AL

ADD DL,48

MOV AH,2H

INT 21H

PRINT MSG4

JMP NXT

NOT\_FOUND: PRINT MSG8

JMP END1

NXT: MOV SI,BX

ADD SI,SI

MOV DI,BX

ADD DI,DI

BACK: SUB SI,02H

CMP SI,00H

JL FORWARD

CMP NUM[SI],CX

JNE FORWARD

MOV BX,SI

SHR BX,1

MOV DL,BL

ADD DL,48

MOV AH,2H

INT 21H

; PRINTMUL BX,JUMP10,JUMP11

PRINT MSG4

JMP BACK

FORWARD:ADD DI,02H

CMP DI,MAX ;MAX has been set to (n - 1)

JG CROSS

CMP NUM[DI],CX

JNE CROSS

MOV BX,DI

SHR BX,1

MOV DL,BL

ADD DL,48

MOV AH,2H

INT 21H

; PRINTMUL BX,JUMP12,JUMP13

PRINT MSG4

JMP FORWARD

CROSS: PRINT MSG6

END1: MOV AH,4CH

INT 21H

MAIN ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL NUM[SI],L5,L6

ADD SI,02H

POP CX

PRINT MSG4

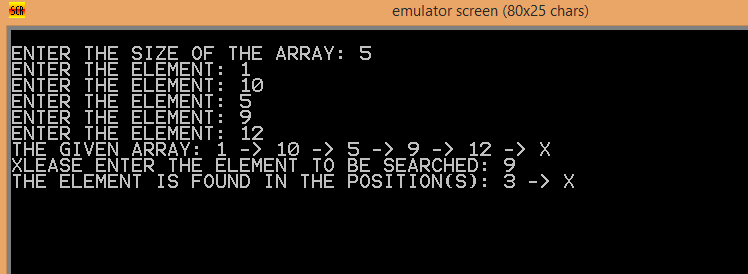
LOOP L4

PRINT MSG6

RET

DISPLAY ENDP

END



A2-5)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

MSG1 DB 10,13,'PLEASE ENTER THE SIZE OF THE ARRAY: $'

MSG2 DB 10,13,'ENTER A NUMBER: $'

MSG3 DB 10,13,'ORIGINAL ARRAY: $'

MSG4 DB ' -> $'

MSG DB 'X$'

MSG5 DB 10,13,'PLEASE ENTER THE KEY TO SEARCHED: $'

MSG6 DB 10,13,'THE GIVEN ELEMENT FOUND IN POSITION(S): $'

N DW 0

NUM DW 100 DUP(0)

KEY DW 0

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG1

READ N,JUMP1,JUMP2

MOV CX,N

MOV SI,0000H

LOOP1: PRINT MSG2

READ NUM[SI],JUMP3,JUMP4

ADD SI,02

LOOP LOOP1

PRINT MSG3

CALL DISPLAY

PRINT MSG5

READ KEY,JUMP5,JUMP6

MOV SI,0000H

MOV CX,N

XOR BL,BL

PRINT MSG6

LOOP2: MOV AX,NUM[SI]

CMP AX,KEY

JNE NEXT

MOV DL,BL

ADD DL,30H

MOV AH,2H

INT 21H

PRINT MSG4

NEXT: INC BL

ADD SI,02H

LOOP LOOP2

PRINT MSG

MOV AH,4CH

INT 21H

MAIN ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL NUM[SI],L5,L6

PRINT MSG4

ADD SI,02H

POP CX

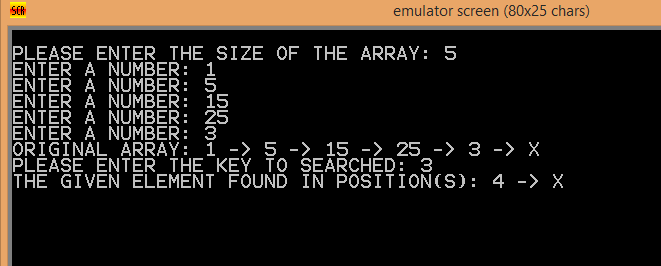
LOOP L4

PRINT MSG

RET

DISPLAY ENDP

END



A2-6)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

NUM DW 100 DUP(0)

N DW 0

CNT1 DW 0

CNT2 DW 0

MSG1 DB 10,13,'PLEASE ENTER THE NO OF ELEMENTS: $'

MSG2 DB 10,13,'ENTER THE ELEMENT: $'

MSG3 DB 10,13,'THE COUNT OF NUMBERS WHICH ARE LESS THAN 20: $'

MSG4 DB ' -> $'

MSG5 DB 10,13,'THE COUNT OF NUMBERS WHICH ARE GREATER THAN 40: $'

MSG6 DB 'X$'

MSG7 DB 10,13,'THE ELEMENTS OF THE ARRAY ARE: $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG1

READ N,JUMP1,JUMP2

MOV CX,N

MOV AX,N

DEC AX

MOV SI,0000H

LOOP1: PRINT MSG2

READ NUM[SI],JUMP3,JUMP4

ADD SI,02H

LOOP LOOP1

PRINT MSG7

CALL DISPLAY

MOV CX,N

MOV SI,0000

LEV: CMP NUM[SI],20

JGE CHK

INC CNT1

JMP NXT

CHK: CMP NUM[SI],40

JLE NXT

INC CNT2

NXT: ADD SI,02H

LOOP LEV

PRINT MSG3

MOV DX,CNT1

ADD DX,30H

MOV AH,2H

INT 21H

PRINT MSG5

MOV DX,CNT2

ADD DX,30H

MOV AH,2H

INT 21H

MOV AH,4CH

INT 21H

MAIN ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL NUM[SI],L5,L6

PRINT MSG4

ADD SI,02H

POP CX

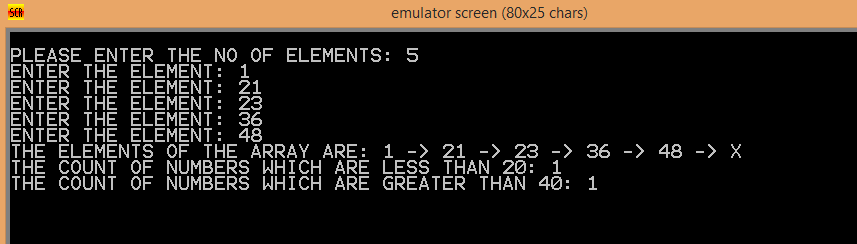
LOOP L4

PRINT MSG6

RET

DISPLAY ENDP

END



A2-7)

.MODEL SMALL

.STACK 32

.DATA

STR1 DB 0DH,0AH, 'ENTER THE BINARY NUMBER :- $'

STR2 DB 0DH,0AH, 'THE HEXADECIMAL EQUIVALENT IS :- $'

STR3 DB 0DH,0AH, 'CURRENT STATE IS: $'

nl db 0dh,0ah,'$'

T1 DB 0

T2 DB 0

T3 DB 0

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

DISP:

LEA DX,STR1

MOV AH,09H

INT 21H

MOV CL,00

READ: MOV AH,01H

INT 21H

MOV BL,AL

PUSH BX

INC CX

CMP AL,0DH

JZ DISPLAY

JMP READ

DISPLAY:

LEA DX,STR2

MOV AH,09H

INT 21H

XOR CX,CX

POP BX

ANS: POP BX

SUB BX,30H

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,1

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,2

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,3

ADD CX,BX

CMP CX,09H

JG LE

ADD CX,30H

JMP RED

LE: ADD CX,30H

ADD CX,7H

RED: MOV T1,CL

XOR CX,CX

XOR BX,BX

POP BX

SUB BX,30H

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,1

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,2

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,3

ADD CX,BX

CMP CX,09H

JG LE1

ADD CX,30H

JMP RED1

LE1: ADD CX,30H

ADD CX,7H

RED1: MOV T2,CL

XOR CX,CX

XOR BX,BX

POP BX

SUB BX,30H

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,1

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,2

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,3

ADD CX,BX

CMP CX,09H

JG LE2

ADD CX,30H

JMP RED2

LE2: ADD CX,30H

ADD CX,7H

RED2: MOV T3,CL

XOR CX,CX

XOR BX,BX

POP BX

SUB BX,30H

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,1

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,2

ADD CX,BX

POP BX

SUB BX,30H

SHL BX,3

ADD CX,BX

CMP CX,09H

JG LE3

ADD CX,30H

JMP RED3

LE3: ADD CX,30H

ADD CX,7H

RED3: MOV DL,CL

MOV AH,2H

INT 21H

MOV DL,T3

MOV AH,2H

INT 21H

MOV DL,T2

MOV AH,2H

INT 21H

MOV DL,T1

MOV AH,2H

INT 21H

END MAIN

END

A2-8)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,9H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

TEMP DW 0

MULT DW 1

MS DB "ENTER THE BINARY NUMBER: $"

MT DB 0DH,0AH,"THE EQUIVALENT DECIMAL NUMBER IS: $"

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MS

XOR CX,CX

LEV:MOV AH,1H

INT 21H

CMP AL,0DH

JE GO

XOR AH,AH

PUSH AX

INC CX

JMP LEV

GO:PRINT MT

OK: POP AX

SUB AX,30H

MUL MULT

MOV BX,TEMP

ADD AX,BX

MOV TEMP,AX ;TEMP=AX\*BX+TEMP

MOV AX,MULT

MOV DX,2H

MUL DX

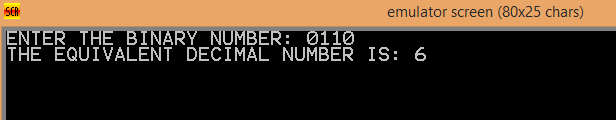
MOV MULT,AX ;MULT=MULT\*16

LOOP OK

PRINTMUL TEMP,B0,V2

MAIN ENDP

END MAIN



A2-9)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

MSG1 DB "ENTER THE NUMBER OF ELEMENTS: $"

MSG2 DB 0DH, 0AH, "ENTER THE ELEMENT: $"

MSG7 DB 0DH, 0AH, "THE CONTENTS OF THE ARRAY ARE: $"

M2 DB 0DH, 0AH, "THE MAXIMUM ELEMENT OF THE ARRAY IS: $"

M5 DB 0DH, 0AH, "AND THE MINIMUM ELEMENT OF THE ARRAY IS: $"

MSG4 DB " -> $"

MSG6 DB "X$"

M45 DB 0DH,0AH,'$'

ARRAY DW 100 DUP(0)

N DW 0

MAXI DW 0

MINI DW 99

UY DW 0

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG1

READ N,JUMP1,JUMP2

MOV CX,N

SHR CX,1

MOV UY,CX

MOV CX,N

MOV SI,0000H

LOOP1: PRINT MSG2

READ ARRAY[SI],JUMP3,JUMP4

ADD SI,02H

LOOP LOOP1

PRINT MSG7

CALL DISPLAY

BV: CALL FIND\_MINIMUM ; call the procedure FIND\_MAXIMUM

CALL FIND\_MAXIMUM ; call the procedure FIND\_MINIMUM

PRINT M45

CALL DISPLAY

MOV AX,00H

MOV MAXI,AX

MOV AX,99H

MOV MINI,AX

MOV AX,UY

DEC AX

JZ FINI

MOV UY,AX

JMP BV

FINI: MOV AH, 4CH

INT 21H

MAIN ENDP

FIND\_MAXIMUM PROC

MOV SI, 00H

MOV CX, N

; MOV AX, ARRAY[SI]

; MOV MAXI, AX

NEXT1: MOV AX, ARRAY[SI]

CMP AX,00H

JE NEXT\_ELEMENT1

CMP MAXI, AX

JG NEXT\_ELEMENT1

MOV MAXI, AX

MOV DI,SI

NEXT\_ELEMENT1:

;INC SI

ADD SI,02H

LOOP NEXT1

LEA DX, M2

MOV AH, 9H

INT 21H

PRINTMUL MAXI,J7,J8

MOV ARRAY[DI],00H

RET

FIND\_MAXIMUM ENDP

FIND\_MINIMUM PROC

MOV SI, 00H

MOV CX, N

;MOV AX, ARRAY[SI]

;MOV MINI, AX

NEXT2: MOV AX, ARRAY[SI]

CMP AX,00H

JE NEXT\_ELEMENT2

CMP MINI, AX

JL NEXT\_ELEMENT2

MOV MINI, AX

MOV DI,SI

NEXT\_ELEMENT2:

ADD SI,02H

LOOP NEXT2

LEA DX, M5

MOV AH, 9H

INT 21H

PRINTMUL MINI,K1,K2

MOV ARRAY[DI],0

RET

FIND\_MINIMUM ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL ARRAY[SI],L5,L6

PRINT MSG4

ADD SI,02H

POP CX

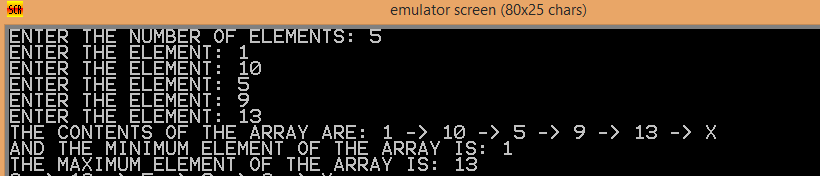
LOOP L4

PRINT MSG6

RET

DISPLAY ENDP

END MAIN



A2-10)

PRINT MACRO MSG

LEA DX,MSG

MOV AH,09H

INT 21H

ENDM

READ MACRO N,J1,J2

J1: MOV AH,01H

INT 21H

CMP AL,0DH

JE J2

SUB AL,30H

MOV BL,AL

MOV AX,N

MOV DX,0AH

MUL DX

XOR BH,BH

ADD AX,BX

MOV N,AX

JMP J1

J2: NOP

ENDM

PRINTMUL MACRO N1,L2,L3

MOV BX,000AH

MOV AX,N1

XOR CX,CX

L2: XOR DX,DX

DIV BX

PUSH DX

INC CX

CMP AX,0000H

JNE L2

L3: POP DX

ADD DL,30H

MOV AH,02H

INT 21H

LOOP L3

ENDM

.MODEL SMALL

.STACK 100H

.DATA

MSG1 DB "ENTER THE NUMBER OF ELEMENTS: $"

MSG2 DB 0DH, 0AH, "ENTER THE ELEMENT: $"

MSG7 DB 0DH, 0AH, "THE CONTENTS OF THE ARRAY ARE: $"

M2 DB 0DH, 0AH, "THE 2ND MAXIMUM ELEMENT OF THE ARRAY IS: $"

M5 DB 0DH, 0AH, "AND THE 2ND MINIMUM ELEMENT OF THE ARRAY IS: $"

MSG4 DB " -> $"

MSG6 DB "X$"

ARRAY DW 100 DUP(0)

N DW 0

MAXI DW 0

MINI DW 99

SEC\_MAXI DW 0

SEC\_MINI DW 99

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

PRINT MSG1

READ N,JUMP1,JUMP2

MOV CX,N

MOV SI,0000H

LOOP1: PRINT MSG2

READ ARRAY[SI],JUMP3,JUMP4

ADD SI,02H

LOOP LOOP1

PRINT MSG7

CALL DISPLAY ; call the procedure PRINT\_ARRAY

CALL FIND\_SECOND\_MAXIMUM ; call the procedure FIND\_SECOND\_MAXIMUM

CALL FIND\_SECOND\_MINIMUM ; call the procedure FIND\_SECOND\_MINIMUM

MOV AH, 4CH

INT 21H

MAIN ENDP

FIND\_SECOND\_MAXIMUM PROC

MOV SI, 00H

MOV CX, N

MOV AX, ARRAY[SI]

MOV MAXI, AX

MOV SEC\_MAXI, AX

NEXT1: MOV AX, ARRAY[SI]

CMP AX, MAXI

JG CHECK

CMP AX,SEC\_MAXI

JG GY

JMP NEXT\_ELEMENT1

CHECK:

;MOV SEC\_MAXI,MAXI

MOV BX,MAXI

MOV SEC\_MAXI,BX

;MOV MAXI,ARRAY[SI]

MOV BX,ARRAY[SI]

MOV MAXI,BX

JMP NEXT\_ELEMENT1

GY: CMP AX,MAXI

JNE HI

JMP NEXT\_ELEMENT1

HI: ;MOV SEC\_MAXI,ARRAY[SI]

MOV BX,ARRAY[SI]

MOV SEC\_MAXI,BX

NEXT\_ELEMENT1:

ADD SI,02H

LOOP NEXT1

LEA DX, M2

MOV AH, 9H

INT 21H

PRINTMUL SEC\_MAXI,K1,K2

;MOV DX, SEC\_MAXI

;ADD DL, 30H

;MOV AH, 2H

;INT 21H

RET

FIND\_SECOND\_MAXIMUM ENDP

FIND\_SECOND\_MINIMUM PROC

MOV SI, 00H

MOV CX, N

;MOV AX, ARRAY[SI]

;MOV MINI, AX

;MOV SEC\_MINI, AX

NEXT2: MOV AX, ARRAY[SI]

CMP AX, MINI

JL CHECK2

CMP AX,SEC\_MINI

JL GY2

JMP NEXT\_ELEMENT2

CHECK2:

;MOV SEC\_MAXI,MAXI

MOV BX,MINI

MOV SEC\_MINI,BX

;MOV MAXI,ARRAY[SI]

MOV BX,ARRAY[SI]

MOV MINI,BX

JMP NEXT\_ELEMENT2

GY2: CMP AX,MINI

JNE HI2

JMP NEXT\_ELEMENT2

HI2: ;MOV SEC\_MAXI,ARRAY[SI]

MOV BX,ARRAY[SI]

MOV SEC\_MINI,BX

NEXT\_ELEMENT2:

ADD SI,02H

LOOP NEXT2

LEA DX, M5

MOV AH, 9H

INT 21H

PRINTMUL SEC\_MINI,G1,G2

RET

FIND\_SECOND\_MINIMUM ENDP

DISPLAY PROC

MOV CX,N

MOV SI,00H

L4: PUSH CX

PRINTMUL ARRAY[SI],L5,L6

PRINT MSG4

ADD SI,02H

POP CX

LOOP L4

PRINT MSG6

RET

DISPLAY ENDP

END MAIN

