**Arrays**

**One Digit Array Input**

.model small

.386

.data

ARRAY DB 10 DUP (?)

DATA1 dw 0000H

msg db 10,13,"Enter the size of the array :: $"

msg2 db 10,13,"Enter the array :: $"

msg3 db 10,13,"The entered array is :: $"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV AH,01

INT 21H

SUB AL,30H

MOV AH,0

MOV CX,AX

MOV DATA1,AX

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

MOV AH,0

MOV SI, 0

MOV BX, OFFSET ARRAY

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV AH, 01H

INT 21H

SUB AL,30H

mov ah, 00

MOV [BX + SI], AX

INC SI

LOOP L1

MOV AH, 09H

MOV DX, OFFSET MSG3

INT 21H

MOV CX, DATA1

MOV SI, 0

L2:mov ah, 02h

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

MOV DX, [BX+SI]

ADD DL, 30h

MOV AH, 02

INT 21H

INC SI

LOOP L2

.EXIT

END

**Two Digits Array Input**

.model small

.386

.data

A1 Db 100 DUP (?)

DATA1 dw 0000H

DATA2 Db 00h

DATA3 Db 00h

msg db 10,13,"Enter the size of the array :: $"

msg2 db 10,13,"Enter the array :: $"

msg3 db 10,13,"The entered array is :: $"

.code

.startup

MOV AH,09

MOV DX, OFFSET msg

INT 21H

MOV CX, 2

L4: MOV AH,01

INT 21H

CMP AL,'A'

JGE L7

SUB AL,30H

JMP L8

L7: SUB AL,37H

L8: SHL BX, 4

ADD BL, AL

LOOP L4

MOV AL, BL

;MOV CL, AL

MOV AH, 0

MOV DATA1, AX

MOV CX, DATA1

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

MOV AH,0

LEA SI, A1

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

Mov DATA2, CL

NUM PROC

MOV CL, 2

AGAIN: MOV AH, 01H

INT 21H

SUB AL,30H

SHL BL,4

ADD BL,AL

LOOP AGAIN

ENDP NUM

MOV CL, DATA2

MOV [SI], BL

INC SI

LOOP L1

MOV AH, 09H

MOV DX, OFFSET MSG3

INT 21H

MOV CX, DATA1

MOV SI, offset A1

L2:mov ah, 02h

mov dl, 0ah

int 21h

MOV DATA3, CL

MOV CL, 2

MOV BL, 00

MOV BL, [SI]

AGAIN2: ROL BL, 4

MOV AL, BL

AND AL, 0FH

MOV DL, AL

ADD DX, 30h

MOV AH, 02

INT 21H

LOOP AGAIN2

MOV CL, DATA3

INC SI

LOOP L2

.EXIT

END

**Addition of two arrays**

.model small

.386

.data

A1 DB 20 DUP (?)

A2 DB 20 DUP (?)

DATA1 dw 0000H

DATA2 DW 0000H

msg db 10,13,"Enter the size of the arrays :: $"

msg2 db 10,13,"Enter the first array :: $"

msg3 db 10,13,"The entered array is :: $

msg4 db 10,13, "Enter the second array ::$"

msg5 db 10,13, "The sum of both array is ::$"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV CX, 2

L4: MOV AH,01

INT 21H

CMP AL,'A'

JGE L9

SUB AL,30H

JMP L8

L9: SUB AL,37H

L8: SHL BX, 4

ADD BL, AL

LOOP L4

MOV AL, BL

MOV CL, AL

MOV AH, 0

MOV DATA1, AX

MOV CX, DATA1

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

;MOV AH,0

MOV CX, DATA1

LEA SI, A1

L1: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV AH, 01H

INT 21H

SUB AL,30H

MOV [SI], AL

INC SI

LOOP L1

MOV AH, 09H

MOV DX, OFFSET MSG3

INT 21H

MOV CX, DATA1

LEA SI, A1

L2:mov ah, 02h

mov dl, 0ah

int 21h

MOV DL, [SI]

ADD DL, 30h

MOV AH, 02

INT 21H

INC SI

LOOP L2

MOV CX, DATA1

MOV AH,09

MOV DX,OFFSET msg4

INT 21H

MOV AH,0

LEA DI, A2

L3: MOV DL, 0AH ; jump onto next line

MOV AH, 02H

INT 21H

MOV AH, 01H

INT 21H

SUB AL,30H

MOV [DI], AL

INC DI

LOOP L3

MOV AH, 09H

MOV DX, OFFSET MSG3

INT 21H

MOV CX, DATA1

LEA DI, A2

L14:mov ah, 02h

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

MOV DX, [DI]

ADD DL, 30h

MOV AH, 02

INT 21H

INC DI

LOOP L14

LEA SI, A1

LEA DI, A2

MOV CX, DATA1

ADDA: MOV AL, [SI]

ADD AL, [DI]

MOV [SI], AL

INC DI

INC SI

LOOP ADDA

MOV AH, 09H

MOV DX, OFFSET MSG5

INT 21H

MOV CX, DATA1

LEA SI, A1

L5:mov ah, 02h

mov dl, 0ah

int 21h

MOV DATA2, CX

MOV CX, 2

MOV BL, [SI]

ADDA1: ROL BL, 4

MOV DL, BL

AND DL, 0FH

CMP Dl, 9

JA L6

ADD DL, 30h

JMP L7

L6: ADD DL, 37H

L7: MOV AH, 02

INT 21H

LOOP ADDA1

MOV CX, DATA2

INC SI

LOOP L5

.EXIT

END

**\*For Subtraction of two arrays, SUB command will be used in place of ADD for subtracting two arrays, everything else will remain same.**