1. **Write a program for 32 - bit binary division and multiplication.**

**Division**

model small

.386

.data

NUM1 dd 00000000H

NUM2 dd 00000000H

REM dd ?

QUO dd ?

msg db 10,13,"Enter the first number here : $"

msg1 db 10,13,"Enter the second number here : $"

msg2 db 10,13,"The Remainder is : $"

msg3 db 10,13,"The Quotient is : $"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV EBX,0

MOV CX,8

AGAIN: MOV AH,01 ; 1ST Nnumber got ENTERED

INT 21H

CMP AL,'A'

JGE L5

JMP L6

L5: SUB AL,37H

L6: SUB AL,30H

SHL EBX,4

ADD BL,AL

LOOP AGAIN

MOV NUM1,EBX

MOV AH,09

MOV DX,OFFSET msg1

INT 21H

MOV EBX,0

MOV CX,8

AGAIN1:MOV AH,01 ;2nd Number got ENTERED

INT 21H

CMP AL,'A'

JGE L7

SUB AL,30H

JMP L8

L7: SUB AL,37H

L8: SHL EBX,4

ADD BL,AL

LOOP AGAIN1

MOV NUM2,EBX

MOV EBX,0

MOV EDX,0

MOV EAX,0

MOV EAX, NUM1

MOV EBX,NUM2

DIV EBX

MOV REM,EDX ; REM = REMAINDER

MOV QUO,EAX ; QUO = QUOTIENT

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

MOV EBX,REM

MOV CX,8

AGAIN2: ROL EBX,4

MOV DL,BL

AND DL,0FH ; to o/p the result in REM

CMP DL,9

JBE L1

ADD DL,37H

MOV AH,02

INT 21H

JMP L2

L1: ADD DL,30H

MOV AH,02

INT 21H

L2: LOOP AGAIN2

MOV AH,09

MOV DX,OFFSET msg3

INT 21H

MOV EBX,QUO

MOV CX,8

AGAIN3: ROL EBX,4

MOV DL,BL

AND DL,0FH ; to o/p the result in Quo

CMP DL,9

JBE L3

ADD DL,37H

MOV AH,02

INT 21H

JMP L4

L3: ADD DL,30H

MOV AH,02

INT 21H

L4: LOOP AGAIN3

.EXIT END

**Multiplication**

.model small

.386

.data

NUM1 dd 00000000H

NUM2 dd 00000000H

PROD1 dd ?

PROD2 dd ?

msg db 10,13,"Enter the first number here : $"

msg1 db 10,13,"Enter the second nnumber here : $"

msg2 db 10,13,"The product(in hexadecimal) is : $"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV EBX,0

MOV CX,8

AGAIN: MOV AH,01 ; 1ST Number got ENTERED

INT 21H

CMP AL,'A'

JGE L5

SUB AL,30H

JMP L6

L5: SUB AL,37H

L6: SHL EBX,4

ADD BL,AL

LOOP AGAIN

MOV NUM1,EBX MOV AH,09

MOV DX,OFFSET msg1 INT 21H

MOV EBX,0 MOV CX,8

AGAIN1:MOV AH,01 ; 2nd number got ENTERED

INT 21H

CMP AL,'A'

JGE L7

SUB AL,30H

JMP L8

L7: SUB AL,37H

L8: SHL EBX,4

ADD BL,AL

LOOP AGAIN1

MOV NUM2,EBX

MOV EBX,0

MOV EDX,0

MOV EAX,0

MOV EAX,NUM1

MOV EBX,NUM2

MUL EBX

MOV PROD1,EDX

MOV PROD2,EAX

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

MOV EBX,PROD1

MOV CX,8

AGAIN2: ROL EBX,4

MOV DL,BL

AND DL,0FH ; to o/p the result

CMP DL,9

JBE L1

ADD DL,37H

MOV AH,02

INT 21H

JMP L2

L1: ADD DL,30H

MOV AH,02

INT 21H

L2: LOOP AGAIN2

MOV EBX,PROD2

MOV CX,8

AGAIN3: ROL EBX,4

MOV DL,BL

AND DL,0FH ; to o/p the result

CMP DL,9

JBE L3

ADD DL,37H

MOV AH,02

INT 21H

JMP L4

L3: ADD DL,30H

MOV AH,02

INT 21H

L4: LOOP AGAIN3

.EXIT

END

1. **Write a program for 32 - bit BCD addition and subtraction.**

**Addition**

.model small

.386

.data

NUM1 DD 00000000H

NUM2 DD 00000000H

NUM3 DD 00000000H

msg db 10,13,"Enter the first number here : $"

msg1 db 10,13,"Enter the second number here : $"

msg2 db 10,13,"The Resultant sum is : $"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV EBX,0

MOV CX,8

AGAIN: MOV AH,01 ; 1ST Number got ENTERED

INT 21H

CMP AL,'A'

JGE L2

SUB AL,30H

SHL EBX,4

ADD BL,AL

LOOP AGAIN

MOV NUM1,EBX

MOV AH,09

MOV DX,OFFSET msg1

INT 21H

MOV EBX,0

MOV CX,8

AGAIN1:MOV AH,01 ; 2nd Number got ENTERED

INT 21H

CMP AL,'A'

JGE L2

SUB AL,30H

SHL EBX,4

ADD BL,AL

LOOP AGAIN1

MOV NUM2, EBX

mov ax, word ptr NUM1

mov dx, word ptr NUM2

add al,dl

daa

mov bl,al

mov al,ah

adc al,dh

daa

mov bh,al

mov word ptr NUM3, bx

mov ax, word ptr NUM1+2

mov dx, word ptr NUM2+2

adc al,dl

daa

mov bl,al

mov al,ah

adc al,dh

daa

mov bh,al

mov word ptr NUM3+2,bx

mov ebx,NUM3

mov ah, 09h

mov dx, offset msg2

int 21h

jnc l6

mov ah, 02h

mov dl, "1"

int 21h

l6: MOV CX,8

AGAIN2: ROL EBX,4

MOV DL,BL

AND DL,0FH

ADD DL,30H

MOV AH,02

INT 21H

LOOP AGAIN2

L2: .EXIT

END

**Subtraction**

.MODEL TINY

.486

.CODE

.STARTUP

; initialize registers

MOV EBX, 37829156H ; BH = 91H BL = 56H

MOV EDX, 45218907H ; DH = 89H DL = 07H

; perform SUBtraction

SUB DL, BL ; DL=07H- 56H = B1H

; perform Decimal Adjust after Subtraction

MOV AL, DL ; AL = B1H

DAS ; AL = 51H C = 1

; store result in ECX(CL)

MOV CL, AL ; CL = 51H ECX = 00000051H

; perform SuBtraction with Borrow

SBB DH, BH ; DH = 89H - 91H - 1 = F7H C = 1

; perform Decimal Adjust after Subtraction

MOV AL, DH ; AL = F7H

DAS ; AL = 97H C = 1

; store result in ECX(CH)

MOV CH, AL ; CH = 97H ECX = 00009751H

; swap EBX, EDX, ECX to perform calculation on higher bits

BSWAP EBX ; EBX = 56918237H BH = 82H BL = 37H

BSWAP EDX ; EDX = B1F72145H DH = 21H DL = 45H

BSWAP ECX ; ECX = 51970000H

; perform SuBtraction with Borrow

SBB DH, BH ; DH = 21H - 82H - 1 = 9EH C = 1

; perform Decimal Adjust after Subtraction

MOV AL, DH ; AL = 9EH

DAS ; AL = (9|3)8H C = 1

; store result in ECX(CH)

MOV CH, AL ; CH = (9|3)8H ECX = 5197(9|3)800H

; perform SuBtraction with Borrow

SBB DL, BL ; DL = 45H - 37H - 1 = 0DH C = 0

; perform Decimal Adjust after Subtraction

MOV AL, DL ; AL = 0DH

DAS ; AL = 07H C = 0

; store result in ECX(CL)

MOV CL,AL ; CL=07H ECX=5197(9|3) 807H

;swap ECX to get final result

BSWAP ECX ; ECX = 07 (9|3) 89751H

.EXIT

END

1. **Write a program for Linear search and binary search.**

**Linear Search**

.model small

.386

.data

ARRAY DW 20 DUP (?)

DATA1 dw 0000H

success db 10,13,"Element is present in the array $"

fail db 10,13,"Element is not present in the arary $"

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

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

msg3 db 10,13,"Enter the element to be searched : $"

.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 DX, SI ; input element of the array

MOV AH, 01H

INT 21H

SUB AL,30H

;MOV SI, DX

MOV [BX + SI], AX

INC SI

LOOP L1

MOV CX,DATA1

MOV AH,09

MOV DX,OFFSET msg3

INT 21H

MOV AH,01 ; Enter element to be searched

INT 21H

SUB AL,30H

MOV SI, 0

MOV BX, OFFSET ARRAY

L2: CMP [BX + SI], AL ; linear search loop

JZ L3 ; jump if element is found

INC SI

LOOP L2

MOV AH, 09H

MOV DX,OFFSET fail ; if the element is not found

INT 21H

MOV AH, 4CH ; to forcefully terminate the program

INT 21H

L3: MOV AH, 09H

MOV DX,OFFSET success ; if the element is found

INT 21H

MOV AH, 4CH

INT 21H

.EXIT END

**Binary Search**

.model small

.386

.data

ARRAY DW 20 DUP (?)

NUM1 dw 0000H

NUM2 dw 0000H

success db 10,13,"Element is present in the array. $"

fail db 10,13,"Element is not present in the arary. $"

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

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

msg3 db 10,13,"Enter the element to be searched in the array : $"

.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 NUM1,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 DX, SI ; input element of the array

MOV AH, 01H

INT 21H

SUB AL,30H

MOV SI, DX

MOV [BX + SI], AX

INC SI

LOOP L1

MOV AH,09

MOV DX,OFFSET msg3

INT 21H

MOV AH,01 ; Enter element to be searched

INT 21H

SUB AL,30H

MOV NUM2,AX

MOV CX,NUM1

MOV SI,0

MOV DI, NUM1

MOV BP, 0

MOV BX, OFFSET ARRAY

MOV AX, NUM1

L2: MOV SI, DI

ADD SI, BP

MOV AX, SI

MOV DL, 2

DIV DL

MOV AH,0

MOV DX,0

MOV SI,AX

MOV DX,NUM2

CMP [BX + SI],DL

JZ L3

CALL L4

LOOP L2

MOV AH, 09H

MOV DX,OFFSET fail ; if the element is not found

INT 21H

MOV AH, 4CH ; to forcefully terminate the program

INT 21H

L3: MOV AH, 09H

MOV DX,OFFSET success ; if the element is found

INT 21H

MOV AH, 4CH

INT 21H

L4 PROC NEAR

CMP [BX+SI], DL

JL L6

MOV DI, SI

RET

L6: MOV BP,SI

RET

L4 ENDP

.EXIT

END

1. **Write a program to add and subtract two arrays.**

**adding two array**

.model small

.386

.data

DATA1 dd 00000000H

msg db 10,13,"Enter the first number : $"

msg1 db 10,13,"Enter the second number : $"

msg2 db 10,13,"The Resultant sum is : $"

.code

.startup MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV EBX,0

MOV CX,8

AGAIN: MOV AH,01 ;1ST NO. ENTERED

INT 21H

CMP AL,'A'

JGE L5

SUB AL,30H

JMP L6

L5: SUB AL,37H

L6: SHL EBX,4

ADD BL,AL

LOOP AGAIN

MOV DATA1,EBX

MOV AH,09

MOV DX,OFFSET msg1

INT 21H

MOV EBX,0

MOV CX,8

AGAIN1:MOV AH,01 ;2nd NO. ENTERED

INT 21H

CMP AL,'A'

JGE L7

SUB AL,30H

JMP L8

L7: SUB AL,37H

L8: SHL EBX,4

ADD BL,AL

LOOP AGAIN1

ADD EBX,DATA1 ;ADDITION

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

MOV CX,8

AGAIN2: ROL EBX,4

MOV DL,BL

AND DL,0FH

CMP DL,09

JG L1 ; to o/p given no.

ADD DL,30H

JMP PRINT

L1: ADD DL,37H

PRINT: MOV AH,02

INT 21H

LOOP AGAIN2

.EXIT

END

**subtracting two array**

.model small

.stack 100H

.data

msg db 10,13,"Enter the first number : $"

msg1 db 10,13,"Enter the second number : $"

msg2 db 10,13,"The Resultant sum is : $"

.code

.startup

MOV AH,09

MOV DX,OFFSET msg

INT 21H

MOV AH, 01

INT 21H

SUB AL,30H

MOV BL, AL

MOV AH,09

MOV DX,OFFSET msg1

INT 21H

MOV AH, 01

INT 21H

SUB AL,30H

SUB BL,AL

MOV AH,09

MOV DX,OFFSET msg2

INT 21H

MOV DL,BL

CMP DL, 09

JG L6

ADD DL,30H

JMP L7

L6: ADD DL, 37H

L7: MOV AH,02

INT 21H

MOV AH, 4CH

INT 21H

.exit

end

1. **binary to ascii.**

.MODEL SMALL

.DATA

INPUT DB 10,13 , 'ENTER BINARY NUMBER : $'

OUTPUT DB 10,13, 'THE ASCII CHARACTER IS : $'

ARR DB ?

.CODE

.STARTUP

MOV AH,09H

MOV DX,OFFSET INPUT

INT 21H

MOV BL, 00H

MOV CL,08H

INPUT1: MOV AH,01H

INT 21H

SUB AL,30H

SHL BL,1

ADD BL,AL

LOOP INPUT1

MOV AH,09H

LEA DX,OUTPUT

INT 21H

MOV AH,02H

MOV DL,BL

INT 21H

.EXIT

END

1. **ascii to binary**

.MODEL SMALL

.DATA

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

RESULT DB 10,13, 'RESULT IS : $'

.CODE

.STARTUP

MOV DX,OFFSET

MESG MOV AH,09H

INT 21H

MOV AH,01H

INT 21H

MOV BL,AL

MOV DX,OFFSET RESULT

MOV AH,09H

INT 21H

MOV CL,08H

MOV AH,00H

MOV AL,BL

L1: SHL AL, 01H

MOV BL,AL

MOV AL,00H

ADC AL,30H

MOV DL,AL

MOV AH,02H

INT 21H

MOV AL,BL

LOOP L1

.EXIT

END