MICROPROCESSOR

PRACTICAL FILE

NAME: Pritam Mandal
COURSE: BSc. (H) Computer Science
ROLL NO.: 21/18043
SUBJECT: Microprocessor

Table of Contents

1. Write a program for 32 bit Addition, Subtraction, Multiplication and Division.

1.1. Addition

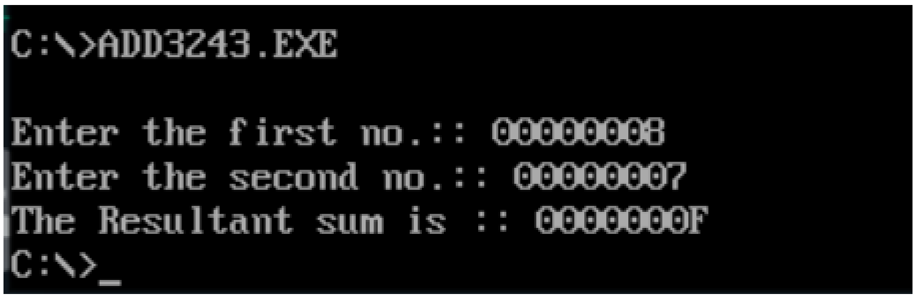
CODE

```
.model small
.stack 100H
.386
.data
data1 dd 00H
msg db 10,13,"Enter the first no.:: $"
msg1 db 10,13,"Enter the second no.:: $"
msg2 db 10,13,"The Resultant sum is :: $"
.code
.startup
   ;initialising EBX with 0
   MOV EBX, 00000000
   ;printing first string
   MOV AH,09
   MOV DX,OFFSET msg
   INT 21H
   ;initialising counter
   MOV ECX, 8
   ;running loop take input of the first  number
   AGAIN: MOV AH, 01
   INT 21H
   CMP AL, 'A'
   JGE P1
   SUB AL,30H
   JMP P4
   P1: SUB AL, 37H
   P4: SHL EBX, 4
   MOV AH,00
   ADD EBX, EAX
   LOOP AGAIN
   ;moving first input to data1
```

```
MOV data1, EBX
;printing second string
MOV AH,09
MOV DX,OFFSET msg1
INT 21H
;setting counter again
MOV ECX, 8
;taking second number as input
AGAIN2: MOV AH, 01
INT 21H
CMP AL, 'A'
JGE P2
SUB AL,30H
JMP P3
P2: SUB AL, 37H
P3: SHL EBX, 4
MOV AH,00
ADD EBX,EAX
LOOP AGAIN2
;adding the two numbers
ADD EBX, data1
;printing output string
MOV AH,09
MOV DX,OFFSET msg2
INT 21H

MOV DX, 00

MOV ECX, 8

AGAIN3: ROL EBX, 4
MOV EDX,EBX
AND DX, 0FH
CMP DX, 09
JG L6
ADD DX,30H
JMP L7
```

```
    L6: ADD DX, 37H
    L7: MOV AH,02
    INT 21H
    LOOP AGAIN3

    MOV AH, 4CH
    INT 21H
  end
```

OUTPUT



```
C:\>ADD3243.EXE

Enter the first no.:: 00000008
Enter the second no.:: 00000007
The Resultant sum is :: 0000000F
C:\>_
```

1.2. Subtraction

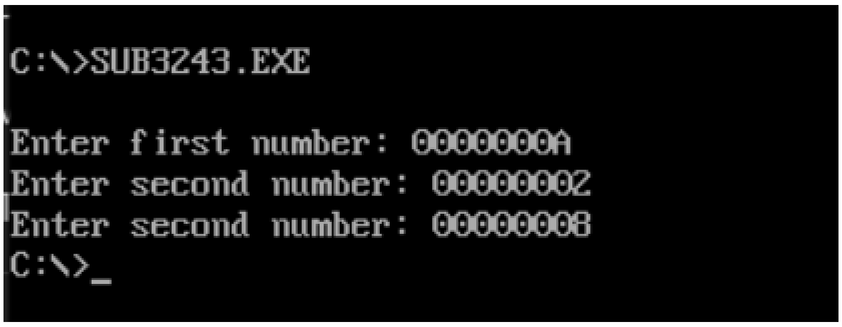
CODE

```
.model small
.stack 100h
.386
.data
data1 dd 00H

str1 db 10,13,"Enter first number: $"
str2 db 10,13,"Enter second number: $"
dif db 10,13,"The difference is: $"

.code
.startup
   mov EBX,00000000

  mov AH,09
  mov DX,offset str1
   int 21h
```

```
mov ECX,8
AGAIN:
    MOV AH,01
    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 str2
int 21h

MOV ECX,8

AGAIN2:
    MOV AH,01
    INT 21H
    CMP AL,'A'
    JGE L7
    SUB AL,30H
    JMP L8
    L7: SUB AL,37H
    L8: SHL EBX,4
    ADD BL,AL
    LOOP AGAIN2

SUB data1,EBX
MOV EBX,data1

mov AH,09
mov DX,offset str2
int 21h
```

```
        MOV ECX,8
        AGAIN3:
            ROL EBX,4
            MOV DL,BL
            AND DL,0FH
            CMP DL,9
            JG L9
            ADD DL,30H
            JMP L10
            L9: ADD DL, 37H
            L10: MOV AH,02
            INT 21H
            LOOP AGAIN3

        MOV AH,4CH
        INT 21H

    end
```

OUTPUT



```
C:\>SUB3243.EXE

Enter first number: 0000000A
Enter second number: 00000002
Enter second number: 00000008
C:\>_
```

1.3. Division

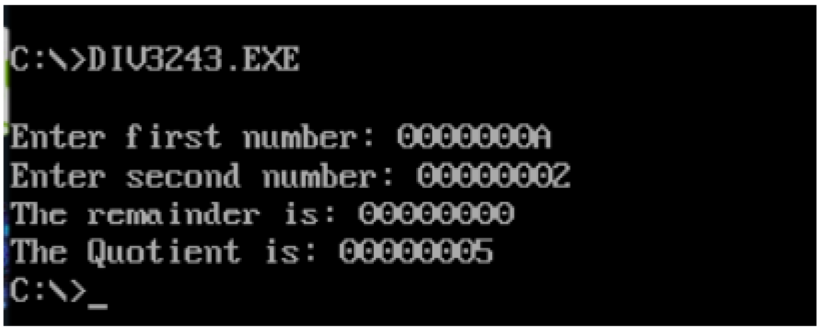
CODE

```
.MODEL SMALL

.STACK 100H
.386
.DATA

REM DD ?
QUO DD ?

DATA1 DD 00000000H
```

```
DATA2 DD 00000000H

str1 db 10,13,"Enter first number: $"
str2 db 10,13,"Enter second number: $"
rem2 db 10,13,"The remainder is: $"
quo2 db 10,13,"The Quotient is: $"

.CODE
.startup

    MOV AH,09
    MOV DX,OFFSET str1
    INT 21H

    MOV EBX,0
    MOV ECX,8
    AGAIN:
        MOV AH,01
        INT 21H

        CMP AL,'A'
        JGE L1
        SUB AL,30H
        JMP L2
        L1: SUB AL,37H
        L2: SHL EBX,4
        ADD BL,AL
        LOOP AGAIN
    MOV DATA1,EBX

    MOV AH,09
    MOV DX,OFFSET str2
    INT 21H

    MOV EBX,0
    MOV ECX,8
    AGAIN2:
        MOV AH,01
        INT 21H

        CMP AL,'A'
        JGE L3
        SUB AL,30H
        JMP L4
        L3: SUB AL,37H
        L4: SHL EBX,4
        ADD BL,AL
        LOOP AGAIN2
    MOV DATA2,EBX

    MOV EBX,0
    MOV EAX,0
    MOV EDX,0
    MOV EAX,DATA1
```

```
        MOV EBX,DATA2
        DIV EBX
        MOV REM,EDX
        MOV QUO,EAX

        MOV AH,09
        MOV DX,OFFSET rem2
        INT 21H

        MOV EBX,REM

        MOV ECX,8
        AGAIN3:
            ROL EBX,4
            MOV DL,BL
            AND DL,0FH
            CMP DL,9
            JB P1
            ADD DL,37H
            JMP P2
            P1:ADD DL,30H
            P2: MOV AH,02
            INT 21H
            LOOP AGAIN3

        MOV AH,09
        MOV DX,OFFSET quo2
        INT 21H
        MOV EBX,QUO

        MOV ECX,8
        AGAIN4:
            ROL EBX,4
            MOV DL,BL
            AND DL,0FH
            CMP DL,9
            JB P3
            ADD DL,37H
            JMP P4
            P3:ADD DL,30H
            P4:MOV AH,02
            INT 21H
            LOOP AGAIN4

        MOV AH,4CH
        INT 21H
    END
```

OUTPUT



```
C:\>DIV3243.EXE

Enter first number: 0000000A
Enter second number: 00000002
The remainder is: 00000000
The Quotient is: 00000005
C:\>_
```

1.4. Multiplication

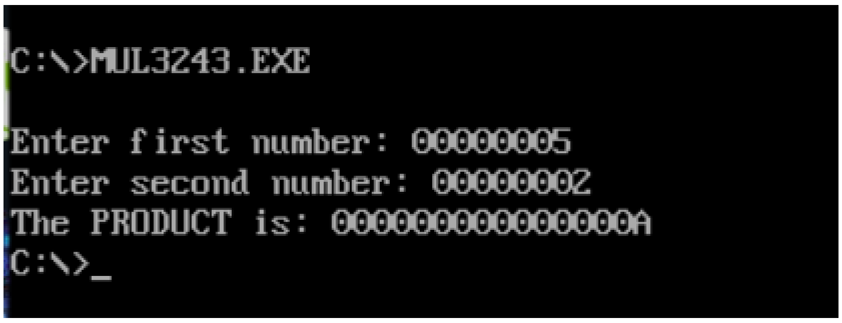
CODE

```
.MODEL SMALL

.STACK 100H
.386
.DATA

PROD1 DD ?
PROD2 DD ?

DATA1 DD 00000000H
DATA2 DD 00000000H

str1 db 10,13,"Enter first number: $"
str2 db 10,13,"Enter second number: $"
PROD db 10,13,"The PRODUCT is: $"

.CODE
.startup

    MOV AH,09
    MOV DX,OFFSET str1
    INT 21H

    MOV EBX,0
    MOV ECX,8
    AGAIN:
        MOV AH,01
        INT 21H

        CMP AL,'A'
        JGE L1
        SUB AL,30H
        JMP L2
        L1: SUB AL,37H
        L2: SHL EBX,4
        ADD BL,AL
        LOOP AGAIN
    MOV DATA1,EBX

    MOV AH,09
    MOV DX,OFFSET str2
```

```
INT 21H

MOV EBX,0
MOV ECX,8
AGAIN2:
    MOV AH,01
    INT 21H

    CMP AL,'A'
    JGE L3
    SUB AL,30H
    JMP L4
    L3: SUB AL,37H
    L4: SHL EBX,4
    ADD BL,AL
    LOOP AGAIN2
MOV DATA2,EBX

MOV EBX,0
MOV EAX,0
MOV EDX,0
MOV EAX,DATA1
MOV EBX,DATA2
MUL EBX
MOV PROD1,EDX
MOV PROD2,EAX

MOV AH,09
MOV DX,OFFSET PROD
INT 21H

MOV EBX,PROD1

MOV ECX,8
AGAIN3:
    ROL EBX,4
    MOV DL,BL
    AND DL,0FH
    CMP DL,9
    JB P1
    ADD DL,37H
    JMP P2
    P1:ADD DL,30H
    P2: MOV AH,02
    INT 21H
    LOOP AGAIN3

MOV EBX,PROD2

MOV ECX,8
AGAIN4:
    ROL EBX,4
    MOV DL,BL
    AND DL,0FH
```

```
                CMP DL,9
                JB P3
                ADD DL,37H
                JMP P4
                P3:ADD DL,30H
                P4:MOV AH,02
                INT 21H
                LOOP AGAIN4

            MOV AH,4CH
            INT 21H
        END
```

OUTPUT



```
C:\>MUL3243.EXE

Enter first number: 00000005
Enter second number: 00000002
The PRODUCT is: 000000000000000A
C:\>_
```

2. Write a program for 32 bit BCD Addition and Subtraction

2.1. Addition

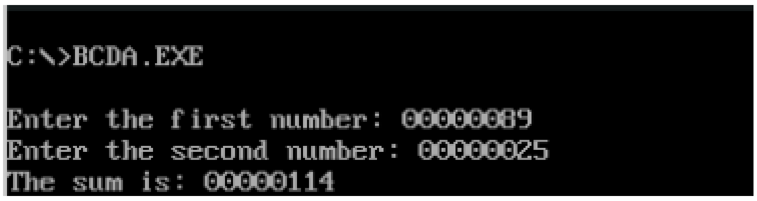
CODE

```
.model small

.stack 100h
.386
.data
    num1 dd 00000000h
    num2 dd 00000000h
    num3 dd 00000000h
    msg db 10,13,"Enter the first number: $"
    msg1 db 10,13,"Enter the second number: $"
    msg2 db 10,13,"The sum is: $"

.code
.startup
    mov ah,09
    mov dx,offset msg
    int 21h

    mov ebx,0
    mov cx,8
    again: mov ah,01
    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
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:mov ah,4ch
        int 21h
    end
```

OUTPUT



```
C:\>BCDA.EXE

Enter the first number: 00000089
Enter the second number: 00000025
The sum is: 00000114
```

2.2. Subtraction

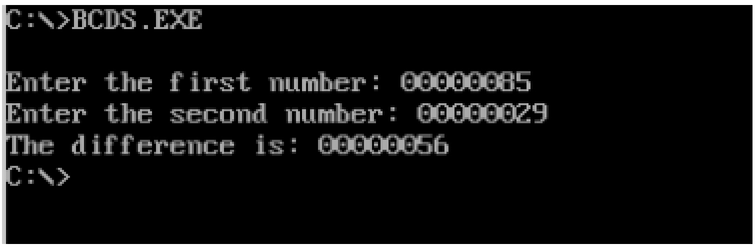
CODE

```
    .model small

    .stack 100h
    .386
    .data
        num1 dd 00000000h
        num2 dd 00000000h
        num3 dd 00000000h
        msg db 10,13,"Enter the first number: $"
        msg1 db 10,13,"Enter the second number: $"
        msg2 db 10,13,"The sum is: $"

    .code
    .startup
        mov ah,09
        mov dx,offset msg
        int 21h

        mov ebx,0
        mov cx,8
        again: mov ah,01
        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
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
sub al,dl
das
mov bl,al
mov al,ah
sbb al,dh
das
mov bh,al
mov word ptr num3,bx
mov ax,word ptr num1+2
mov dx,word ptr num2+2
sbb al,dl
das
mov bl,al
mov al,ah
sbb al,dh
das
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:mov ah,4ch
          int 21h
      end
```

OUTPUT



```
C:\>BCDS.EXE

Enter the first number: 00000085
Enter the second number: 00000029
The difference is: 00000056
C:\>
```

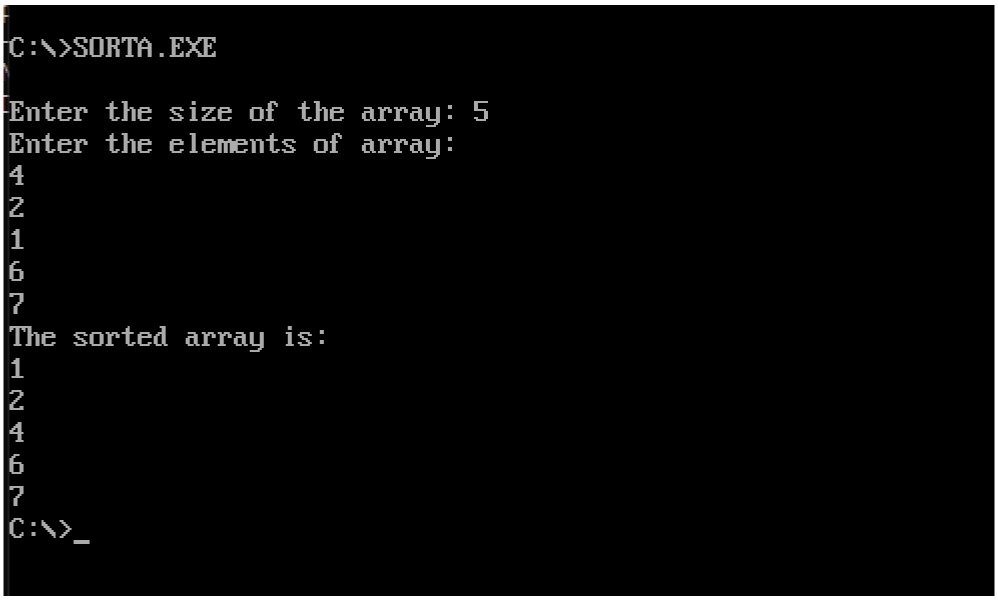
3. Write a program for sorting.

CODE

```
.model SMALL

.stack 100H
.386
.data
ARRAY dw 20 DUP (?)
DATA1 dw 0000H
NUMB dw 0000H
msg db 10,13,"Enter the size of the array: $"
msg2 db 10,13,"Enter the elements of array: $"
msg3 db 10,13,"The sorted array is: $"

.code
.startup

mov ah, 09h
mov dx, offset msg
int 21h

mov ah, 01h
int 21h

sub al, 30h
mov ah, 0
mov cx, ax
mov DATA1, ax
mov ah, 09h
mov dx, offset msg2
int 21h

mov ah, 0
mov si, 0
mov bx, offset ARRAY

l1: mov dl, 0ah
    mov ah, 02h
    int 21h
    mov dx, si
    mov ah, 01h
    int 21h
    sub al, 30h
    mov si, dx
    mov [bx + si], ax
    inc si
loop l1

mov cx, DATA1
mov bx, offset ARRAY
mov di, cx

l2: mov cx, DATA1
    mov NUMB, cx
    dec NUMB
    mov cx, NUMB
    mov si, 0

l3: mov al, [bx + si]
    cmp [bx + si + 1], al
    jl l4
    xchg al, [bx + si + 1]
    mov [bx + si], al

l4: inc si

loop l3

dec di
jnz l2

MOV CX, DATA1
MOV SI, DATA1
dec SI
MOV BX, OFFSET ARRAY

MOV AH,09
MOV DX, OFFSET msg3
INT 21H

l5: mov dl, 0ah
    mov ah, 02h
    int 21h

    mov dx, [bx + si]
    dec si
    add dl, 30h

    mov ah, 02
    int 21h

loop l5
```

```
mov ah, 4ch
int 21h
end
```

OUTPUT



```
C:\>SORTA.EXE

Enter the size of the array: 5
Enter the elements of array:
4
2
1
6
7
The sorted array is:
1
2
4
6
7
C:\>_
```

4. Write a program for Linear Search and Binary Search.

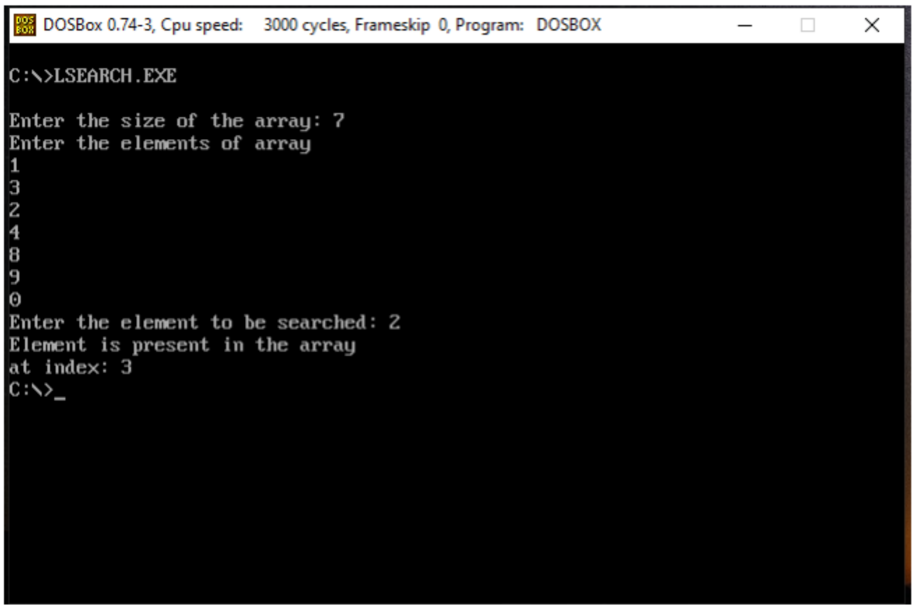
4.1. Linear Search

CODE

```
.model small

.stack 100h
.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 array $"

    msg db 10,13,"Enter the size of the array: $"
    msg2 db 10,13,"Enter the elements of array$"
    msg3 db 10,13,"Enter the element to be searched: $"
    msg4 db 10,13,"at index: $"
.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
    mov ah,02h
    int 21h

    mov dx,si

    mov ah,01h
    int 21h

    sub al,30h

    mov [bx+si],ax
    inc si
    loop L1

mov cx,DATA1
mov ah,09
mov dx,offset msg3
int 21h

mov ah,01
int 21h
sub al,30h

mov si,0
mov bx,offset ARRAY

L2:
    cmp [bx+si],al
    jz L3
    inc si
    loop L2
mov ah,09
mov dx,offset fail
int 21h

mov ah,4ch
int 21h

L3:
    mov ah,09
    mov dx,offset success
    int 21h
```

```
        mov ah,09
        mov dx,offset msg4
        int 21h

        inc si
        mov dx,si
        add dx,30h
        mov ah,02
        int 21h

        mov ah,4ch
        int 21h
end
```

OUTPUT


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
C:\>LSEARCH.EXE

Enter the size of the array: 7
Enter the elements of array
1
3
2
4
8
9
0
Enter the element to be searched: 2
Element is present in the array
at index: 3
C:\>_
```

4.2. Binary Search

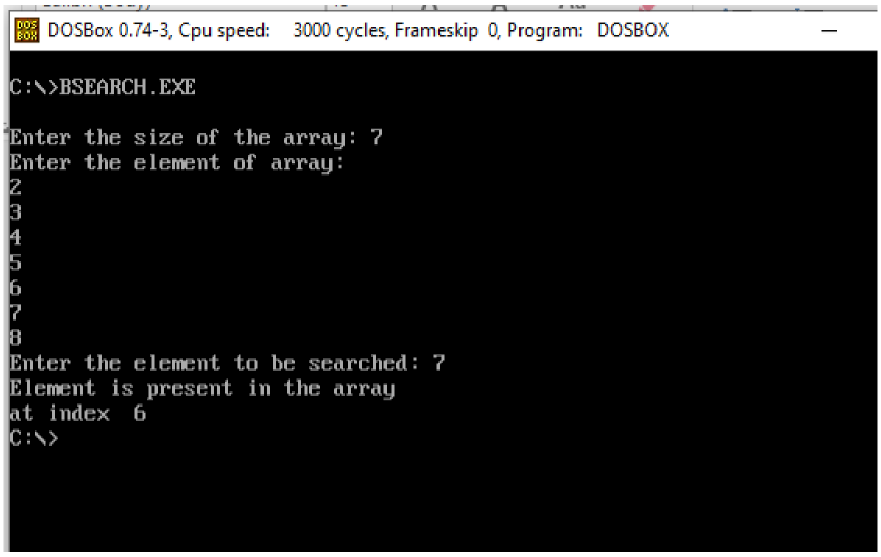
CODE

```
.model small

.stack 100h
.386
.data
        ARRAY dw 20 DUP (?)
        DATA1 dw 0000h
        DATA2 dw 0000h
        success db 10,13,"Element is present in the array $"
        fail db 10,13,"Element is not present in the array $"

        msg db 10,13,"Enter the size of the array: $"
        msg2 db 10,13,"Enter the element of array: $"
        msg3 db 10,13,"Enter the element to be searched: $"
        msg4 db 10,13,"at index  $"

.code
.startup
```

```
    mov ah,09h
    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
            mov ah,02h
            int 21h

            mov dx,si
            mov ah,01h
            int 21h

            sub al,30h

            mov [bx+si],ax
            inc si
            loop L1




    mov ah,09
    mov dx,offset msg3
    int 21h

    mov ah,01
    int 21h
    sub al,30h

MOV DATA2,AX
MOV CX,DATA1
MOV SI,0
MOV DI, DATA1
MOV BP, 0
MOV BX, OFFSET ARRAY
MOV AX, DATA1
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,DATA2
            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, 09H
            MOV DX,OFFSET msg4
            INT 21H

            MOV DX,SI
            ADD DX,30H
            ADD DX,01

            MOV AH, 02
            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

            MOV AH, 4CH
            INT 21H
    END
```

OUTPUT



```
C:\>BSEARCH.EXE

Enter the size of the array: 7
Enter the element of array:
2
3
4
5
6
7
8
Enter the element to be searched: 7
Element is present in the array
at index  6
C:\>
```

5. Write a program to add and subtract two arrays.

5.1. Addition

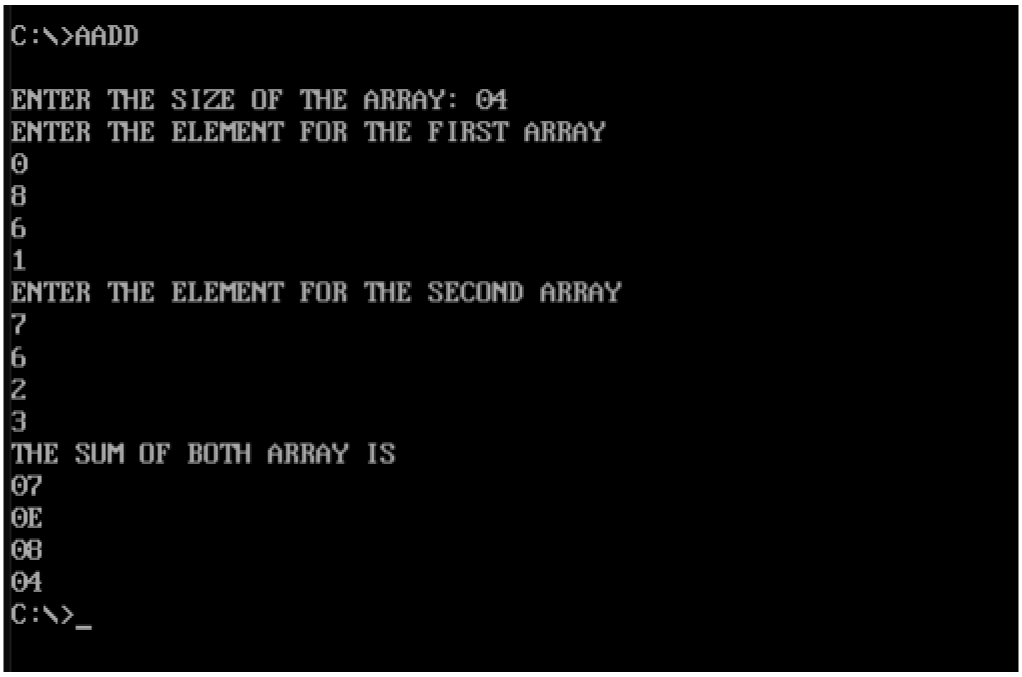
CODE

```
.model small

.stack 100h
.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 ARRAY: $"
    MSG2 DB 10,13,"ENTER THE ELEMENT FOR THE FIRST ARRAY $"
    MSG3 DB 10,13,"ENTER THE ELEMENT FOR THE SECOND ARRAY $"
    MSG4 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 CX, DATA1
MOV AH,09
MOV DX,OFFSET msg3
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

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 MSG4
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

    MOV AH,4CH
    INT 21H
END
```

OUTPUT



```
C:\>AADD

ENTER THE SIZE OF THE ARRAY: 04
ENTER THE ELEMENT FOR THE FIRST ARRAY
0
8
6
1
ENTER THE ELEMENT FOR THE SECOND ARRAY
7
6
2
3
THE SUM OF BOTH ARRAY IS
07
0E
08
04
C:\>_
```

5.2. Subtraction

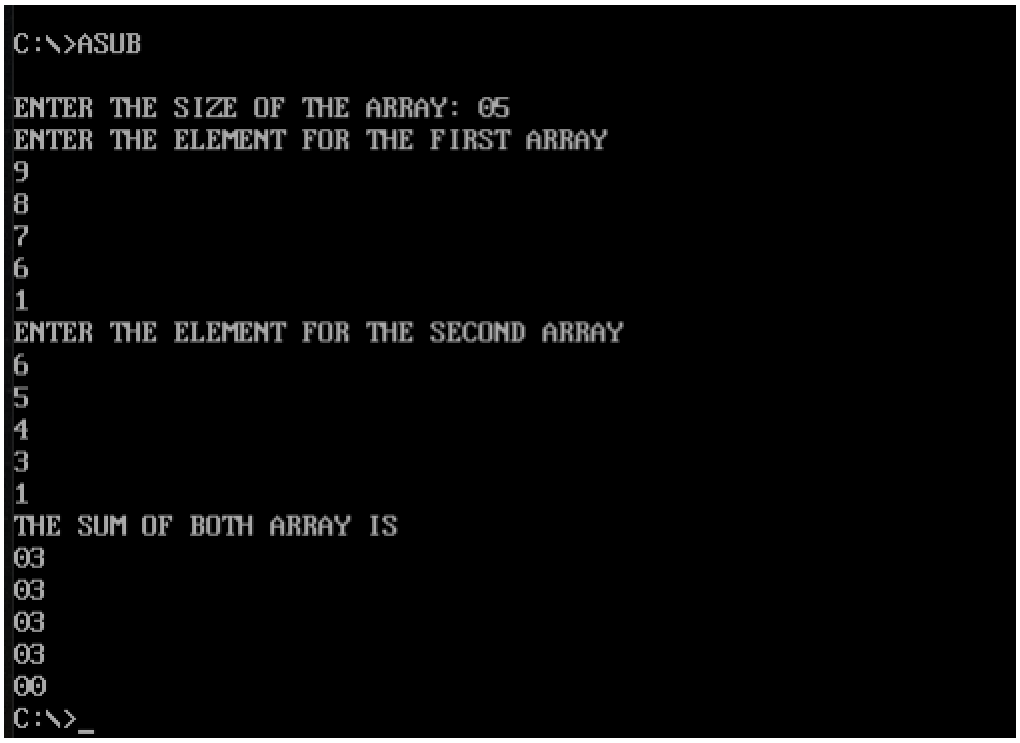
CODE

```
.model small

.stack 100h
.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 ARRAY: $"
    MSG2 DB 10,13,"ENTER THE ELEMENT FOR THE FIRST ARRAY $"
    MSG3 DB 10,13,"ENTER THE ELEMENT FOR THE SECOND ARRAY $"
    MSG4 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 CX, DATA1
    MOV AH,09
    MOV DX,OFFSET msg3
```

```
        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

        LEA SI, A1
        LEA DI, A2
        MOV CX, DATA1
        ADDA: MOV AL, [SI]
        SUB AL, [DI]
        MOV [SI], AL
        INC DI
        INC SI
        LOOP ADDA

        MOV AH, 09H
        MOV DX, OFFSET MSG4
        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

        MOV AH,4CH
        INT 21H
    END
```

OUTPUT



```
C:\>ASUB

ENTER THE SIZE OF THE ARRAY: 05
ENTER THE ELEMENT FOR THE FIRST ARRAY
9
8
7
6
1
ENTER THE ELEMENT FOR THE SECOND ARRAY
6
5
4
3
1
THE SUM OF BOTH ARRAY IS
03
03
03
03
00
C:\>_
```

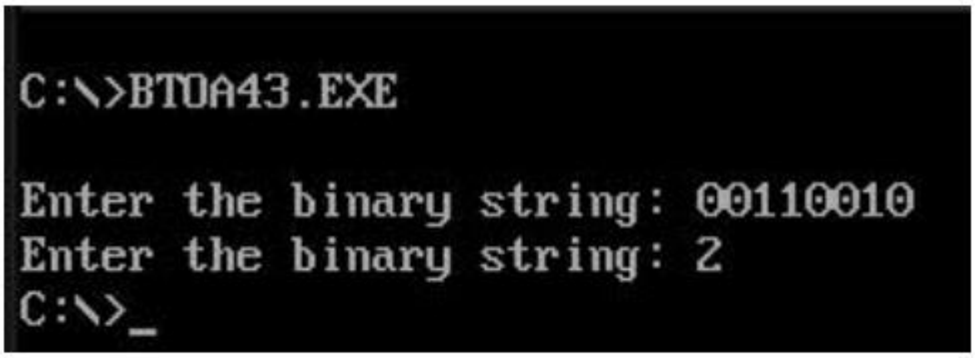
6. Write a program to perform binary to ascii conversion.

CODE

```
.model small

.stack 100h
.data
        msg db 10,13,"Enter the binary character: $"
        msg2 db 10,13,"The ASCII character is: $"
.code
.startup
        mov ah,09h
        mov dx,offset msg
        int 21h

        mov bl,0
        mov cl,8
        L1:
                mov ah,01h
                int 21h
                shl bl,1
                sub al,30h
                add bl,al
                LOOP L1

        mov ah,09h
        mov dx, offset msg2
        int 21h

        mov ah,02h
        mov dl,bl
        int 21h
```

```
            mov ah,4ch
            int 21h
    end
```

OUTPUT



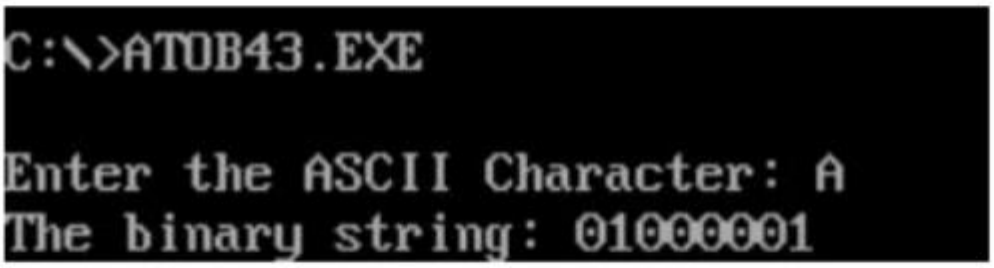
```
C:\>BTOA43.EXE

Enter the binary string: 00110010
Enter the binary string: 2
C:\>_
```

7. Write a program to perform ascii to binary conversion.

CODE

```
.model SMALL

.stack 100H
.data
    inputStr db 10,13, 'Enter an ASCII Character: $'
    outputStr db 10,13, 'Binary equivalent is : $'
.code
.startup

        ; print the input string
    MOV DX, OFFSET inputStr
    MOV AH, 09H
    INT 21H

    MOV AH, 01H
    INT 21H

    MOV BL, AL

    ; print the output string
    MOV DX, OFFSET outputStr
    MOV AH, 09H
    INT 21H

    MOV CX, 8

    repeat8Times:
        SHL BL, 1
        JC printOne

        MOV DL, 30H
        JMP print

        printOne:
            MOV DL, 31H

        print:
            MOV AH, 02H
            INT 21H
        LOOP repeat8Times


    MOV AH, 4CH
        INT 21H

END
```

OUTPUT



```
C:\>ATOB43.EXE

Enter the ASCII Character: A
The binary string: 01000001
```