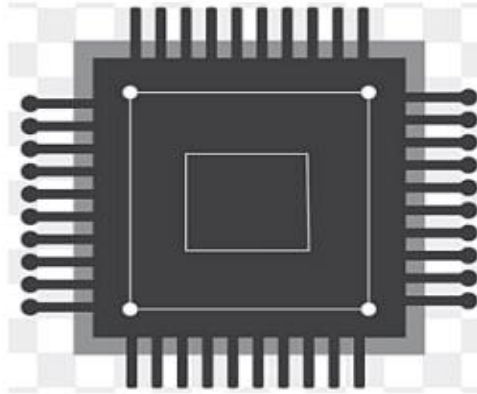




RAM LAL ANAND COLLEGE
UNIVERSITY OF DELHI



MICROPROCESSOR

PRATICAL FILE FOR PAPER CODE 32347504

Submitted By :

Harsh Jaiswal

Examination Roll no. 20058570011

BSc (Hons) Computer Science

Submitted To :

Dr. Vandana Gandotra

Department of Computer Science

Question1 Write a program for 32-bit binary Addition ,Subtraction, Division ,and Multiplication .

Solution:

32-Bit Binary Addition:

```
.model small
```

```
.386
```

```
.data
```

```
DATA1 dd 00000000H
```

```
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
```

```
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

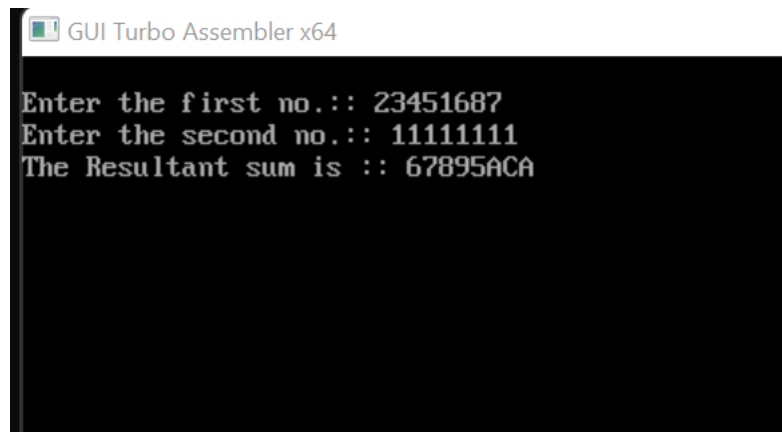
MOV DL,BL

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  
  
END
```

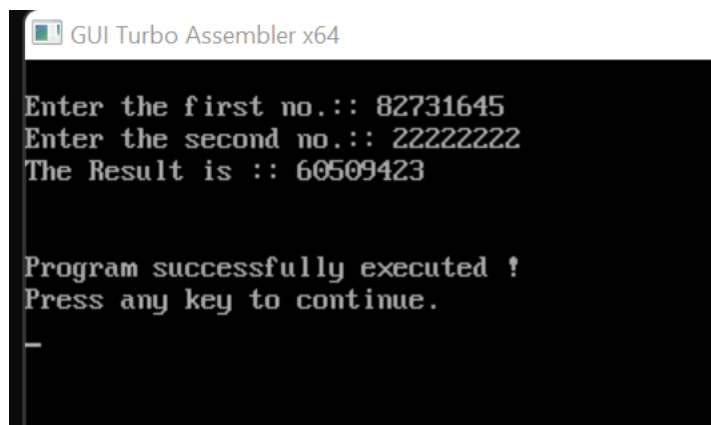


32-Bit Binary Subtraction:-

```
.model small  
  
.386  
  
.data  
DATA1 dd 00000000H  
msg db 10,13,"Enter the first no.:: $"  
msg1 db 10,13,"Enter the second no.:: $"  
msg2 db 10,13,"The Result 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
SUB 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
END
```



32-bit Multiplication:-

```
.model small
```

```
.386
```

```
.data
```

```
DATA1 dd 00000000H
```

DATA2 dd 00000000H

PROD1 dd ?

PROD2 dd ?

msg db 10,13,"Enter the first no.:: \$"

msg1 db 10,13,"Enter the second no.:: \$"

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 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

MOV DATA2,EBX

MOV EBX,0

MOV EDX,0

MOV EAX,0

MOV EAX,DATA1

MOV EBX,DATA2

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
```

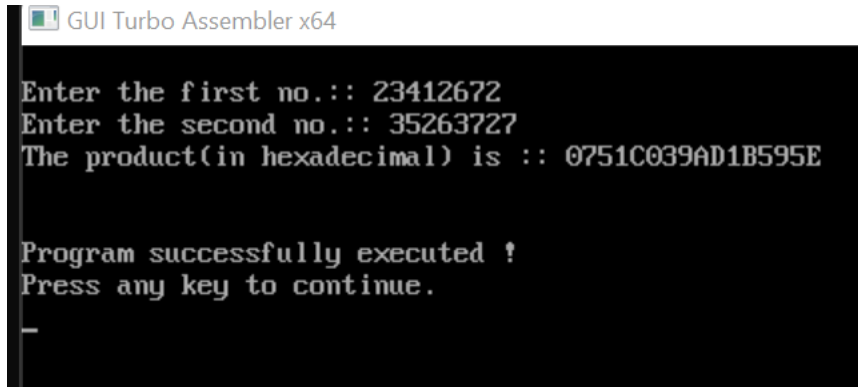
```
L2: LOOP AGAIN2
MOV EBX,PROD2
MOV CX,8
AGAIN3: ROL EBX,4
MOV DL,BL
AND DL,0FH ; to o/p the result
CMP DL,9
JBE L3
ADD DL,37H
MOV AH,02
INT 21H
JMP L4
```

```
L3: ADD DL,30H
MOV AH,02
INT 21H
L4: LOOP AGAIN3
```

MOV AH,4CH

INT 21H

END

A screenshot of the GUI Turbo Assembler x64 window. The window has a black background with white text. The text displays the execution of a program that takes two numbers as input and outputs their product in hexadecimal. The input numbers are 23412672 and 35263727, and the resulting product is 0751C039AD1B595E. The program execution is successful, and the user is prompted to press any key to continue.

```
GUI Turbo Assembler x64
Enter the first no.: 23412672
Enter the second no.: 35263727
The product(in hexadecimal) is :: 0751C039AD1B595E

Program successfully executed !
Press any key to continue.
_
```

32-bit division

model small

.386

.data

DATA1 dd 00000000H

DATA2 dd 00000000H

REM dd ?

QUO dd ?

msg db 10,13,"Enter the first no.: \$"

msg1 db 10,13,"Enter the second no.: \$"

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 NO. 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 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

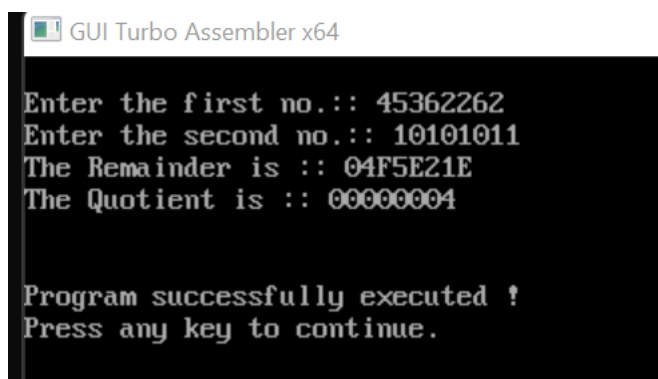
JMP L8

L7: SUB AL,37H

```
L8: SHL EBX,4
ADD BL,AL
LOOP AGAIN1
MOV DATA2,EBX
MOV EBX,0
MOV EDX,0
MOV EAX,0
MOV EAX,DATA1
MOV EBX,DATA2
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
MOV AH,4CH
INT 21H
END
```



GUI Turbo Assembler x64

```
Enter the first no.: 45362262
Enter the second no.: 10101011
The Remainder is :: 04F5E21E
The Quotient is :: 00000004

Program successfully executed !
Press any key to continue.
```

Question 2 :-Write a program for 32-Bit BCD Addition and Subtraction.

Solution:-

32-Bit BCD Addition

.MODEL SMALL

.386

.DATA

MESS0 DB 10,13,"ENTER THE FIRST NUMBER:\$"

MESS1 DB 10,13,"ENTER THE SECOND NUMBER:\$"

MESS2 DB 10,13,"THE SUM IS:\$"

A DD ?

B DD ?

C DD ?

COUNT DB 04h

.CODE

.STARTUP

LEA DX,MESS0

MOV AH,09

INT 21H

MOV EBX,0

MOV CX,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 A,EBX

LEA DX,MESS1

MOV AH,09

INT 21H

MOV EBX,0

MOV CX,8

AGAIN:

MOV AH,01

INT 21H

CMP AL,'A'

JGE L51

SUB AL,30H

JMP L61

L51: SUB AL,37H

L61: SHL EBX,4

ADD BL,AL

LOOP AGAIN

MOV B,EBX

MOV AX,WORD PTR A

MOV BX,WORD PTR B

ADD AL,BL

DAA

MOV BL,AL

ADC AH,BH

MOV AL,AH

DAA

MOV BH,AL

MOV WORD PTR C,BX

MOV AX,WORD PTR A+2

MOV BX,WORD PTR B+2

ADC AL,BL

DAA

MOV BL,AL

ADC AH,BH

MOV AL,AH

DAA

MOV BH,AL

MOV WORD PTR C+2,BX

LEA DX,MESS2

MOV AH,09

INT 21H

MOV BX,WORD PTR C+2

MOV DH,2

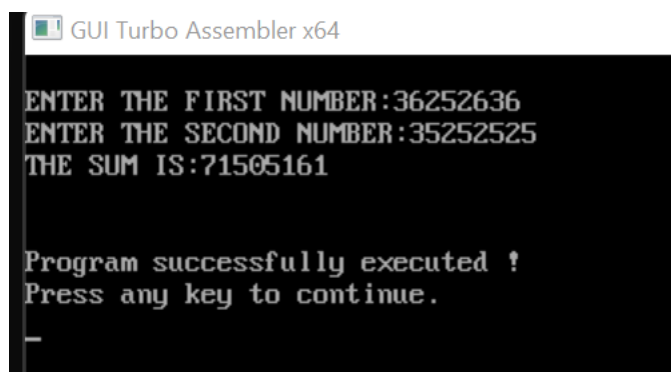
L1: MOV CH,04H

MOV CL,04H


```
L2: ROL BX,CL
MOV DL,BL
AND DL,0FH
CMP DL,09
JBE L4
ADD DL,07

L4: ADD DL,30H
MOV AH,02
INT 21H
DEC CH
JNZ L2
DEC DH
CMP DH,0
MOV BX,WORD PTR C
JNZ L1

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



32-Bit BCD SUBTRATION:-

.MODEL SMALL

.386

.DATA

MESS0 DB 10,13,"ENTER THE FIRST NUMBER:\$"

MESS1 DB 10,13,"ENTER THE SECOND NUMBER:\$"

MESS2 DB 10,13,"THE DIFFERENCE IS:\$"

A DD ?

B DD ?

C DD ?

COUNT DB 04h

.CODE

.STARTUP

LEA DX,MESS0

MOV AH,09

INT 21H

MOV EBX,0

MOV CX,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 A,EBX

LEA DX,MESS1

MOV AH,09

INT 21H

MOV EBX,0

MOV CX,8

AGAIN:

MOV AH,01

INT 21H

CMP AL,'A'

JGE L51

SUB AL,30H

JMP L61

L51: SUB AL,37H

L61: SHL EBX,4

ADD BL,AL

LOOP AGAIN

MOV B,EBX

MOV AX,WORD PTR A

MOV BX,WORD PTR B

SUB AL,BL

DAS

MOV BL,AL

SBB AH,BH

MOV AL,AH

DAS

MOV BH,AL

MOV WORD PTR C,BX

MOV AX,WORD PTR A+2

MOV BX,WORD PTR B+2

SBB AL,BL

DAS

MOV BL,AL

SBB AH,BH

MOV AL,AH

DAS

MOV BH,AL

MOV WORD PTR C+2,BX

LEA DX,MESS2

MOV AH,09

INT 21H

MOV BX,WORD PTR C+2

MOV DH,2

L1: MOV CH,04H

MOV CL,04H

L2: ROL BX,CL

MOV DL,BL

AND DL,0FH

CMP DL,09

JBE L4

ADD DL,07

L4: ADD DL,30H

MOV AH,02

INT 21H

DEC CH

JNZ L2

DEC DH

CMP DH,0

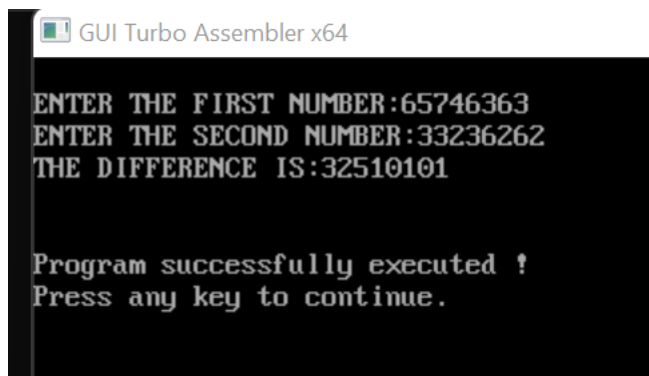
MOV BX,WORD PTR C

JNZ L1

MOV AH,4CH

INT 21H

END



Question3:- Write a program for Sorting.

Solution:-

.model small

.386

.data

ARRAY DW 20 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 sorted 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 DX, SI ; input element of the array
```

```
MOV AH, 01H
```

```
INT 21H
```

```
SUB AL,30H
```

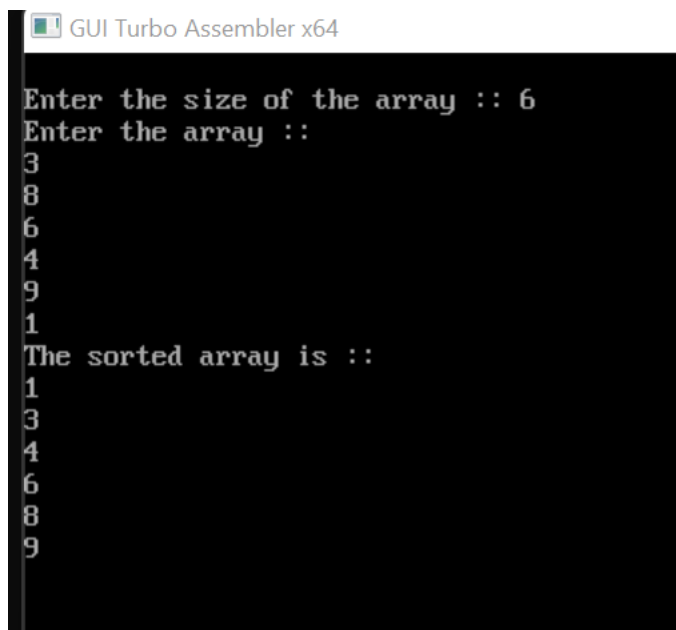
```
MOV SI, DX
MOV [BX + SI], AX
INC SI
LOOP L1
MOV CX, DATA1
MOV BX, OFFSET ARRAY
MOV DI,CX
L2: MOV CX, DATA1
MOV SI, 0

L3: MOV AL, [BX + SI]
CMP AL, [BX + SI + 1]
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, 1
MOV BX, OFFSET ARRAY
MOV AH,09
MOV DX,OFFSET msg3
INT 21H

L5: MOV DL, 0AH ; jump onto next line
MOV AH, 02H
INT 21H
MOV DX, [BX + SI]
```

```
INC SI  
ADD DL, 30H  
MOV AH, 02  
INT 21H  
LOOP L5  
  
END
```



```
GUI Turbo Assembler x64  
Enter the size of the array :: 6  
Enter the array ::  
3  
8  
6  
4  
9  
1  
The sorted array is ::  
1  
3  
4  
6  
8  
9
```

Question4:- Write a program for Linear search and Binary Search.

Solution:-

Linear – Search:-

```
.MODEL SMALL  
.STACK  
.386  
.DATA
```

```
ARRAY DB 9 DUP(?)
```

```
MESS01 DB 13,10,"MAX. NO. OF ELEMENTS IN ARRAY IS 9 $"
```

```
MESS02 DB 13,10," $"
```

```
MESS1 DB 13,10,"ENTER THE NUMBER OF ELEMENTS: $"
```


MESS0 DB 13,10,"ENTER THE NUMBER: \$"

MESS2 DB 13,10,"ENTER THE ELEMENT TO BE SEARCHED: \$"

MESS3 DB 13,10,"VALUE FOUND AT LOCATION - \$"

MESS4 DB 13,10,"VALUE NOT FOUND!!!\$"

ErrMess DB 13,10,"ERROR IN INPUT DIGIT\$"

DAT DB ?

number dw ?

POS DW ?

.CODE

.STARTUP

MOV DX, OFFSET MESS01

MOV AH, 09

INT 21H

MOV DX, OFFSET MESS02

MOV AH, 09

INT 21H

MOV DX,OFFSET MESS1

MOV AH, 09

INT 21H

MOV AH, 01

INT 21H

CMP al,39h

JBE abc

MOV DX, OFFSET ErrMess

MOV AH, 09

INT 21H

JMP myexit

abc: AND AL, 0FH

MOV AH, 0

MOV number, AX

MOV CX, AX ; SET COUNTER AL TIMES

MOV DI, 0

; INPUT ELEMENTS IN ARRAY

MYLOOP:

MOV DX, OFFSET MESS0

MOV AH, 09

INT 21H

; Tens digit

MOV AH, 01

INT 21H

CMP AL, 39H

JBE abc2

MOV DX, OFFSET ErrMess

MOV AH,09

INT 21H

JMP myexit

abc2: AND al,0fh

SHL AL, 4

MOV BL, AL

; Units digit

MOV AH,01

INT 21H

cmp al,39h

jbe abcx

MOV DX,OFFSET ErrMess

MOV AH,09

INT 21H

jmp myexit

abcx:

AND al,0fh

ADD al, bl

MOV ARRAY[DI], AL

INC DI

LOOP MYLOOP

;INPUT ELEMENT TO BE SEARCHED

MOV DX,OFFSET MESS2

MOV AH,09

INT 21H

; Tens digit

MOV AH,01

INT 21H

cmp al,39h

jbe abcl

MOV DX,OFFSET ErrMess

MOV AH,09

INT 21H

jmp myexit

abcl:

and al,0fh

shl al,4

mov bl,al

; Units digit

MOV AH,01

INT 21H

cmp al,39h

jbe abcm

MOV DX,OFFSET ErrMess

MOV AH,09

INT 21H

jmp myexit

abcm:

and al,0fh

add al,bl

mov DAT,AL

; SEARCH PROCESS

MOV AX, DS

MOV ES, AX

MOV AL, DAT

CLD ; Auto-Increment Mode

MOV CX, number

MOV DI, OFFSET ARRAY

REPNE SCASB

CMP CX, 0

JE NOTFOUND

MOV DX, OFFSET MESS02

MOV AH, 09

INT 21H

MOV DX, OFFSET MESS3

MOV AH,09

INT 21H

SUB NUMBER, CX

ADD NUMBER,30H

MOV DX, NUMBER

MOV AH, 02

INT 21H

JMP myexit

NOTFOUND:

MOV DX,OFFSET MESS4

MOV AH,09

INT 21H

myexit:

MOV DX, OFFSET MESS02

MOV AH, 09

INT 21H

.EXIT

END

```
MAX. NO. OF ELEMENTS IN ARRAY IS 9

ENTER THE NUMBER OF ELEMENTS: 5
ENTER THE NUMBER: 64
ENTER THE NUMBER: 83
ENTER THE NUMBER: 38
ENTER THE NUMBER: 86
ENTER THE NUMBER: 74
ENTER THE ELEMENT TO BE SEARCHED: 38

VALUE FOUND AT LOCATION - 3

Program successfully executed !
Press any key to continue.
```

Binary-Search :-

.model small

.stack

.386

.data

ARRAY DB 10 DUP(?)

MESS0 DB 13,10,"ENTER THE NUMBER: \$"

MESS1 DB 13,10,"ENTER THE NUMBER OF ELEMENTS: \$"

MESS2 DB 13,10,"ENTER THE ELEMENT TO BE SEARCHED: \$"

MESS3 DB 13,10,"VALUE FOUND AT LOCATION- \$"

MESS4 DB 13,10,"VALUE NOT FOUND!!!\$"

ErrMess DB 13,10,"ERROR IN INPUT DIGIT\$"

DAT DB ?

number dw ?

.code

.startup

MOV DX,OFFSET MESS1

MOV AH,09

INT 21H

MOV AH,01

INT 21H

cmp al,39h

jbe abc

MOV DX,OFFSET ErrMess

MOV AH,09

INT 21H

jmp myexit

abc:

and al,0fh

mov ah,0

mov number,ax

MOV CX,AX

MOV DI,0

MYLOOP:

MOV DX,OFFSET MESS0

MOV AH,09

INT 21H

MOV AH,01

INT 21H

cmp al,39h

jbe abc2

MOV DX,OFFSET ErrMess

MOV AH,09

INT 21H

jmp myexit

abc2:

and al,0fh

MOV ARRAY[DI],AL

INC DI

LOOP MYLOOP

MOV DX,OFFSET MESS2

MOV AH,09

INT 21H

MOV AH,01

INT 21H

cmp al,39h

jbe abc3

MOV DX,OFFSET ErrMess

MOV AH,09

INT 21H

jmp myexit

abc3:

and al,0fh

MOV DAT,AL

mov ax,ds

mov es,ax

mov al,dat

CLD

mov cx,number

INC CX

mov DI, offset ARRAY

repne SCASB

CMP CX,0

JE NTFOUND

MOV DX,OFFSET MESS3

MOV AH,09

INT 21H

SUB NUMBER,CX ;FIND ELEMENT LOCATION

ADD NUMBER,30H

MOV DX,NUMBER

INC DX

MOV AH,02

INT 21H

JMP myexit

NTFOUND:

MOV DX,OFFSET MESS4

MOV AH,09

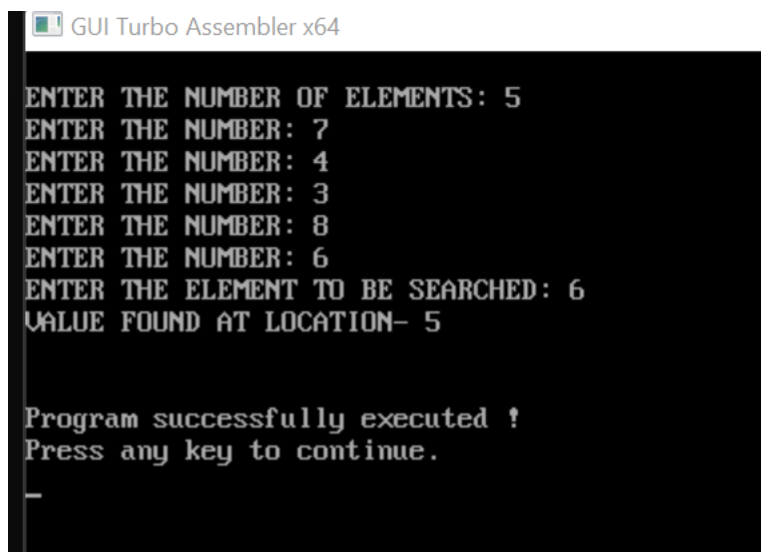
INT 21H

myexit:

MOV AH,4CH

INT 21H

END

A screenshot of the GUI Turbo Assembler x64 window. The window has a black background with white text. The title bar reads "GUI Turbo Assembler x64". The main area displays the following text:

```
ENTER THE NUMBER OF ELEMENTS: 5
ENTER THE NUMBER: 7
ENTER THE NUMBER: 4
ENTER THE NUMBER: 3
ENTER THE NUMBER: 8
ENTER THE NUMBER: 6
ENTER THE ELEMENT TO BE SEARCHED: 6
VALUE FOUND AT LOCATION- 5

Program successfully executed !
Press any key to continue.
```

Question5:- Write a program to add and subtract two array.

Solution:-

Addition of Two Array:

.model small

.data

mat1 db 12h, 11h, 12h, 10h, 11h, 12h, 10h, 11h, 12h

mat2 db 13h, 02h, 02h, 02h, 02h, 02h, 02h, 02h, 02h

res3 dw 9 dup(?)

.code

mov ax, @data

mov ds, ax

mov cx, 09h

mov di, offset mat1

mov bx, offset mat2

mov si, offset res3

back : mov ah, 0

mov al, [di]

add al, [bx]

adc ah, 00

mov [si], ax

inc di

inc bx

inc si

inc si

loop back

mov si, offset res3

mov dh, 9

l10: mov ch, 04h

mov cl, 04h

mov bx, [si]

l2: rol bx, cl

mov dl, bl

and dl, 0fh

cmp dl, 09

jbe l4

add dl, 07

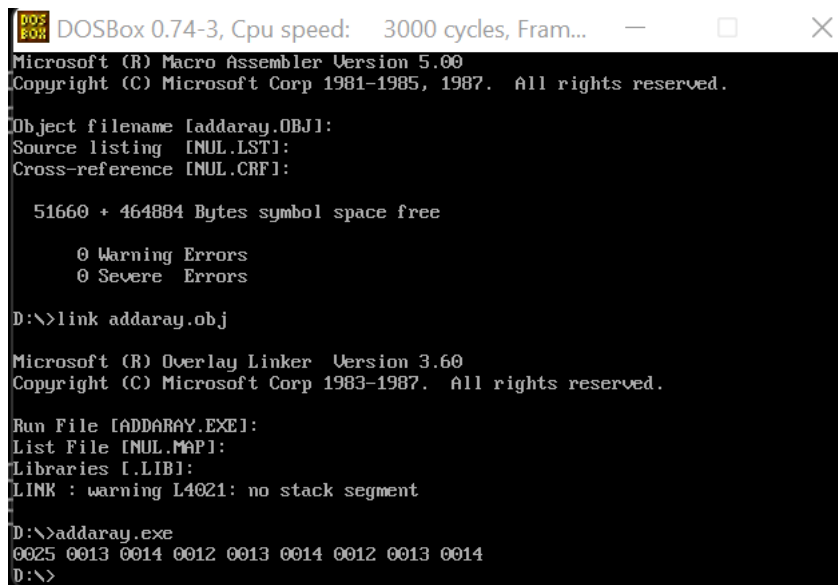
l4: add dl, 30h

mov ah, 02

int 21h

dec ch

```
jnz l2  
  
mov dl, ' ' ;This is a whitespace  
  
int 21h  
  
inc si  
  
inc si  
  
dec dh  
  
jnz l10  
  
mov ah, 4ch  
  
int 21h  
  
  
end
```



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...  
Microsoft (R) Macro Assembler Version 5.00  
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.  
Object filename [addarray.OBJ]:  
Source listing [NUL.LST]:  
Cross-reference [NUL.CRF]:  
  
51660 + 464884 Bytes symbol space free  
  
0 Warning Errors  
0 Severe Errors  
  
D:\>link addarray.obj  
  
Microsoft (R) Overlay Linker Version 3.60  
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.  
  
Run File [ADDARRAY.EXE]:  
List File [NUL.MAP]:  
Libraries [.LIB]:  
LINK : warning L4021: no stack segment  
  
D:\>addarray.exe  
0025 0013 0014 0012 0013 0014 0012 0013 0014  
D:\>
```

Subtraction of two array:-

```
.model small  
  
.data  
  
mat1 db 12h, 11h, 12h, 10h, 11h, 12h, 10h, 11h, 12h  
  
mat2 db 13h, 02h, 02h, 02h, 02h, 02h, 02h, 02h, 02h  
  
res3 dw 9 dup(?)  
  
.code
```

```
mov ax, @data
mov ds, ax
mov cx, 09h
mov di, offset mat1
mov bx, offset mat2
mov si, offset res3
back : mov ah, 0
mov al, [di]
sub al, [bx]
adc ah, 00
mov [si], ax
inc di
inc bx
inc si
inc si
loop back
mov si, offset res3
mov dh, 9
```

```
l10: mov ch, 04h
mov cl, 04h
mov bx, [si]
```

```
l2: rol bx, cl
mov dl, bl
and dl, 0fh
cmp dl, 09
jbe l4
add dl, 07
```

```
l4: add dl, 30h

mov ah, 02

int 21h

dec ch

jnz l2

mov dl, ' ' ;This is a whitespace

int 21h

inc si

inc si

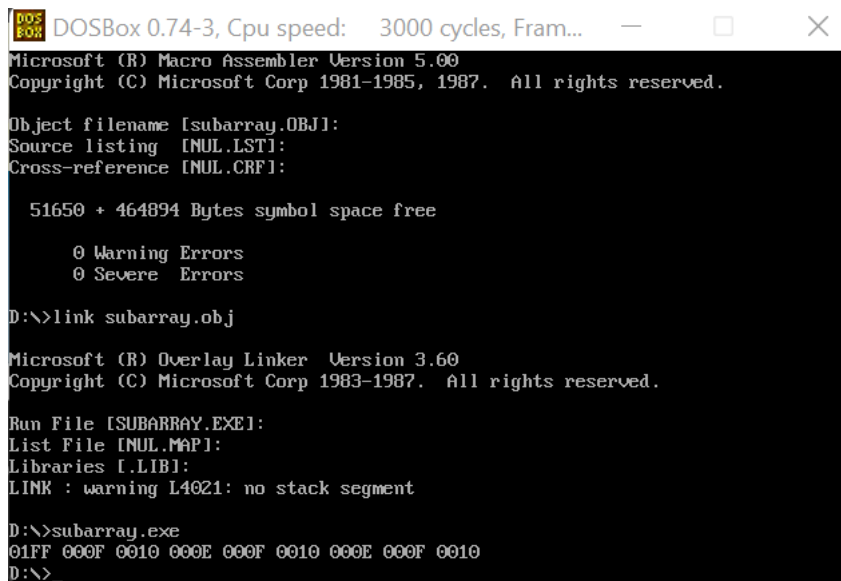
dec dh

jnz l10

mov ah, 4ch

int 21h

end
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

```
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [subarray.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51650 + 464894 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\>link subarray.obj

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [SUBARRAY.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:
LINK : warning L4021: no stack segment

D:\>subarray.exe
01FF 000F 0010 000E 000F 0010 000E 000F 0010
D:\>_
```

Question6:-write a program for binary to ascii conversion.

.MODEL SMALL

.DATA

INPUT DB 10,13 , 'ENTER BINARY NO: \$'

```
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,
```

```
INT 21H
```

```
MOV AH,02H
```


```
MOV DL,BL
```

```
INT 21H
```

```
MOV AH,4CH
```

```
INT 21H
```

```
END
```

 GUI Turbo Assembler x64

```
ENTER BINARY NO: 10100010
THE ASCII CHARACTER IS:ó
```

```
Program successfully executed !
Press any key to continue.
```

Question 7:-Write a program for ascii to binary conversion.

```
.model small
```

```
.stack 100h
```

```
.data
```

```
input db "Enter an ASCII character :$"
```

```
db 10,13,"Binary Equiv: $"
```

```
.code
```

```
MOV AX ,@DATA
```

```
MOV DS ,AX
```

```
MOV DX ,OFFSET input
```

```
MOV AH, 09H
```

```
INT 21H
```

```
MOV AH,01H
```

```
INT 21H
```

```
MOV BL,AL
```

```
MOV DX,OFFSET
```

```
MOV AH,09H
```

```
INT 21H
```

```
MOV CX,8
```

```
BIN_EQUIV:
```

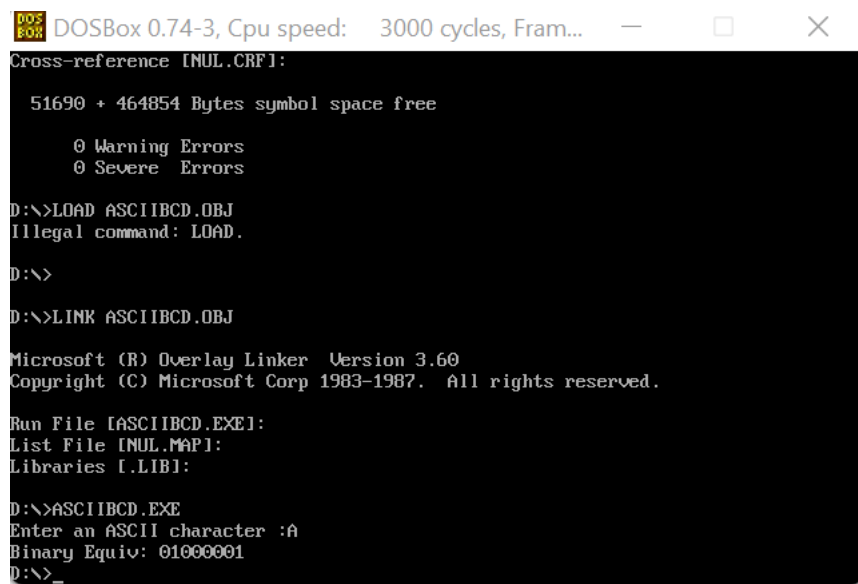


```
SHL BL,1
JC PRINTONE
PRINTZERO:
    MOV DL,30H
    JMP PRINT
PRINTONE:
    MOV DL,31H
PRINT:
    MOV AH,02H
    INT 21H
    LOOP BIN_EQUIV
```

```
MOV AH,4CH
```

```
INT 21H
```

```
END
```



DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

```
Cross-reference INUL.CRF1:
51690 + 464854 Bytes symbol space free
0 Warning Errors
0 Severe Errors
D:\>LOAD ASCIIBCD.OBJ
Illegal command: LOAD.
D:\>
D:\>LINK ASCIIBCD.OBJ
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [ASCIIBCD.EXE]:
List File [INUL.MAP]:
Libraries [.LIB]:
D:\>ASCIIBCD.EXE
Enter an ASCII character :A
Binary Equiv: 01000001
D:\>_
```