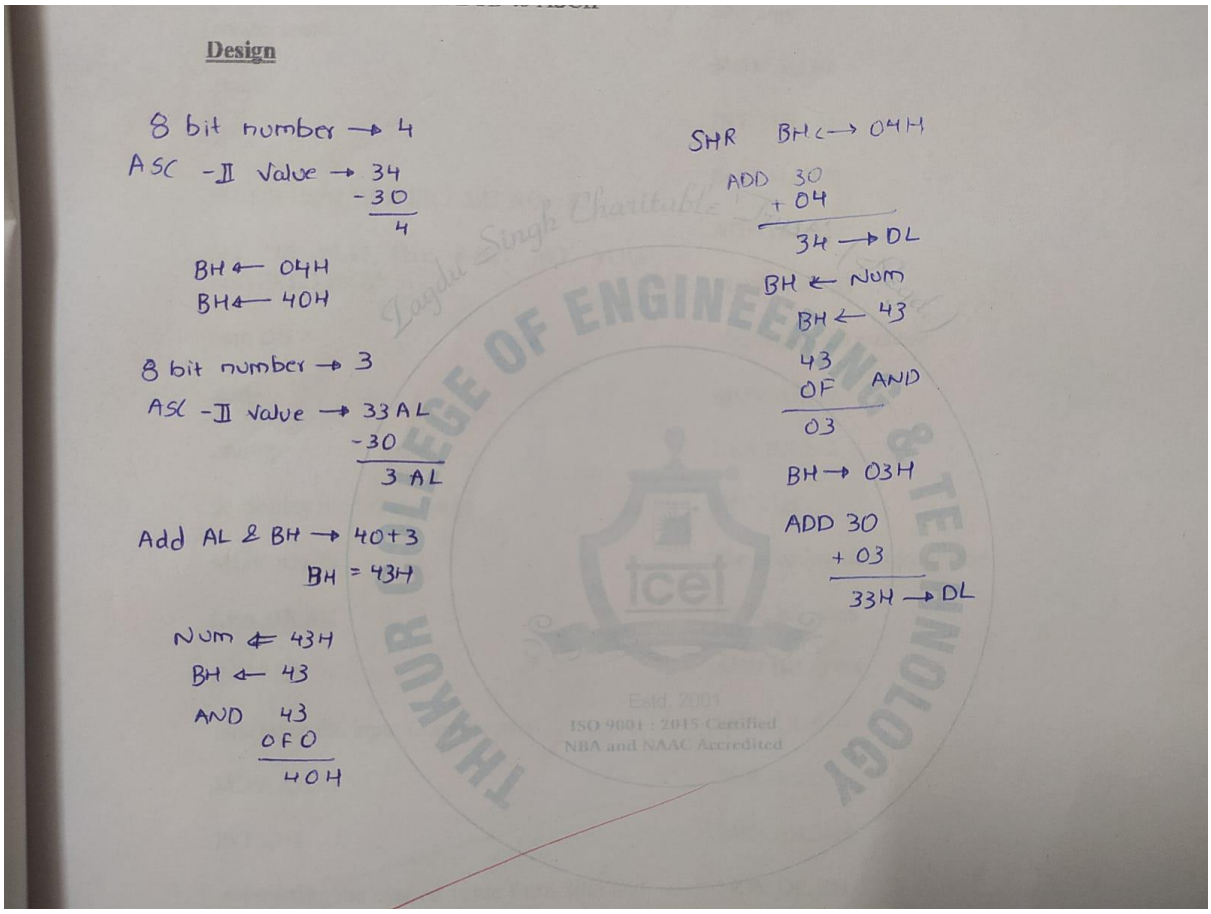
Display 8bit number



Design

8 bit number → 4
ASC -II value → 34
-30
4

BH ← 04H
BH ← 40H

8 bit number → 3
ASC -II value → 33 AL
-30
3 AL

Add AL & BH → 40+3
BH = 43H

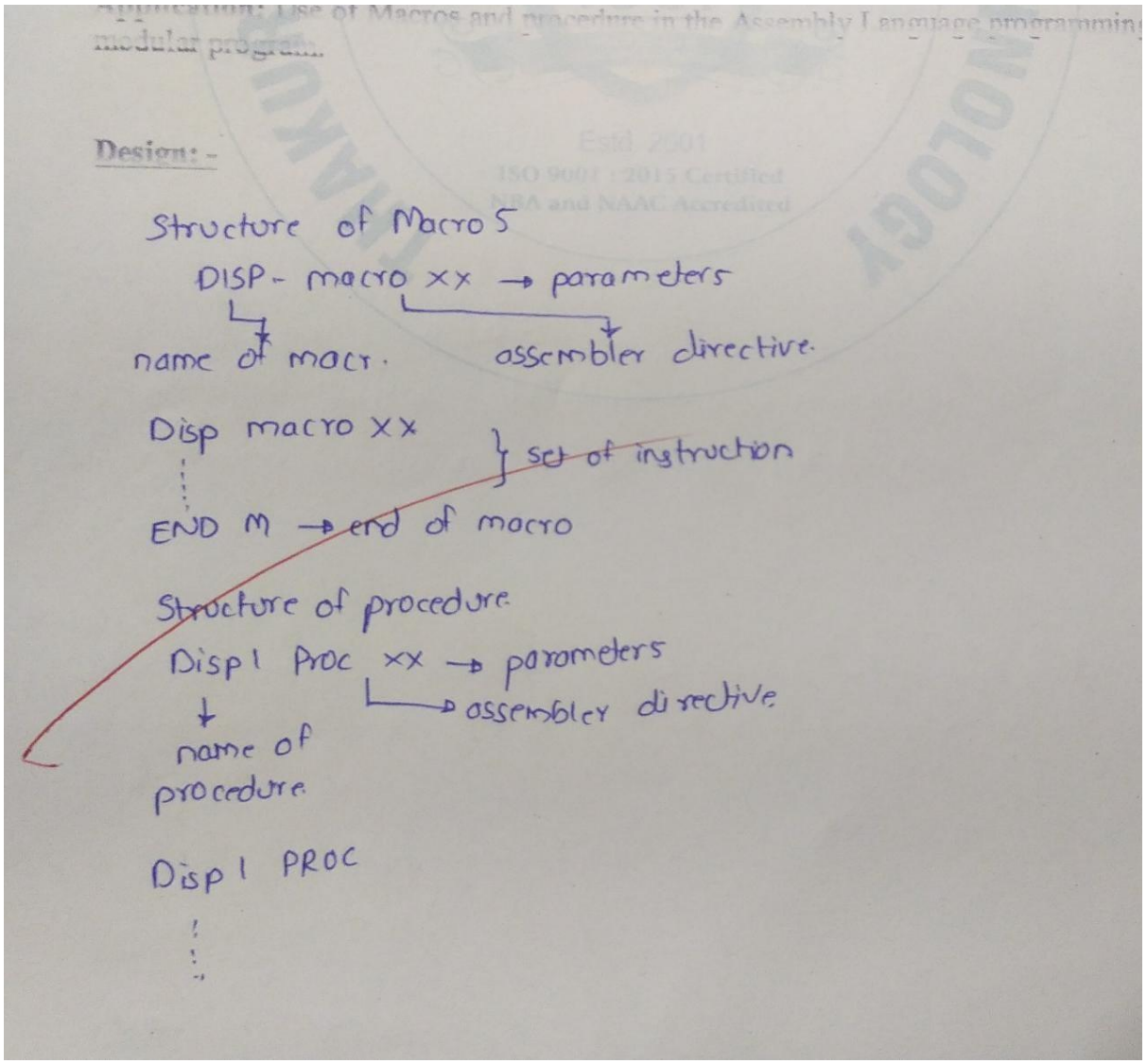
Num ← 43H
BH ← 43
AND 43
0F 0
40H

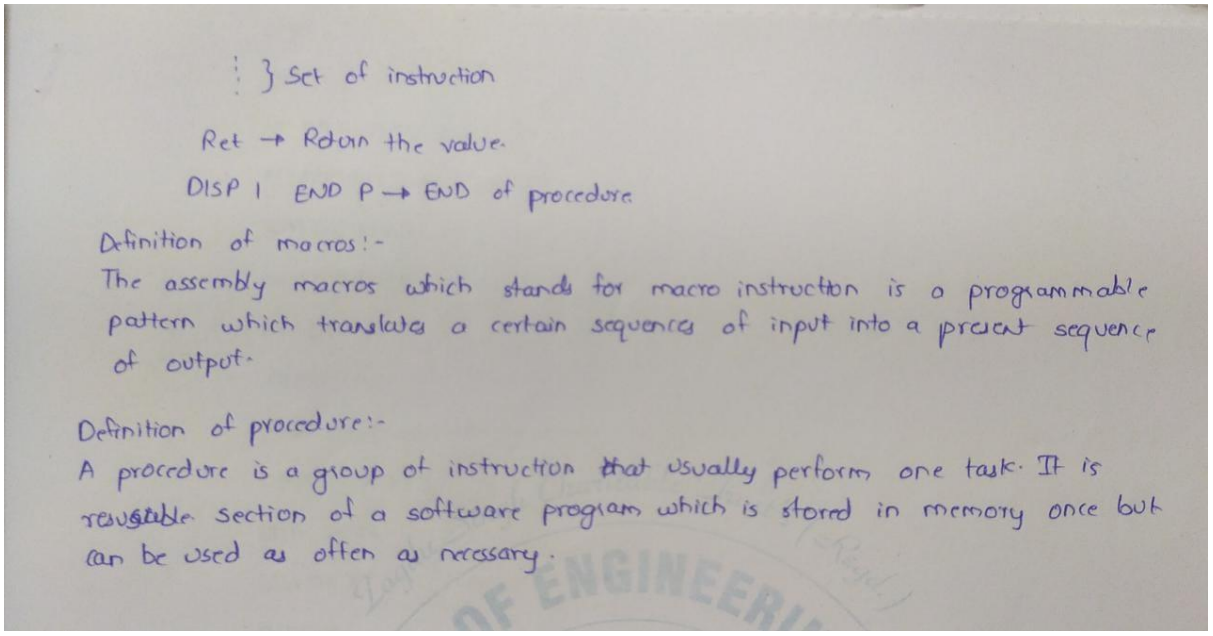
SHR BH ← 04H
ADD 30
+04
34 → DL

BH ← Num
BH ← 43
43
0F AND
03

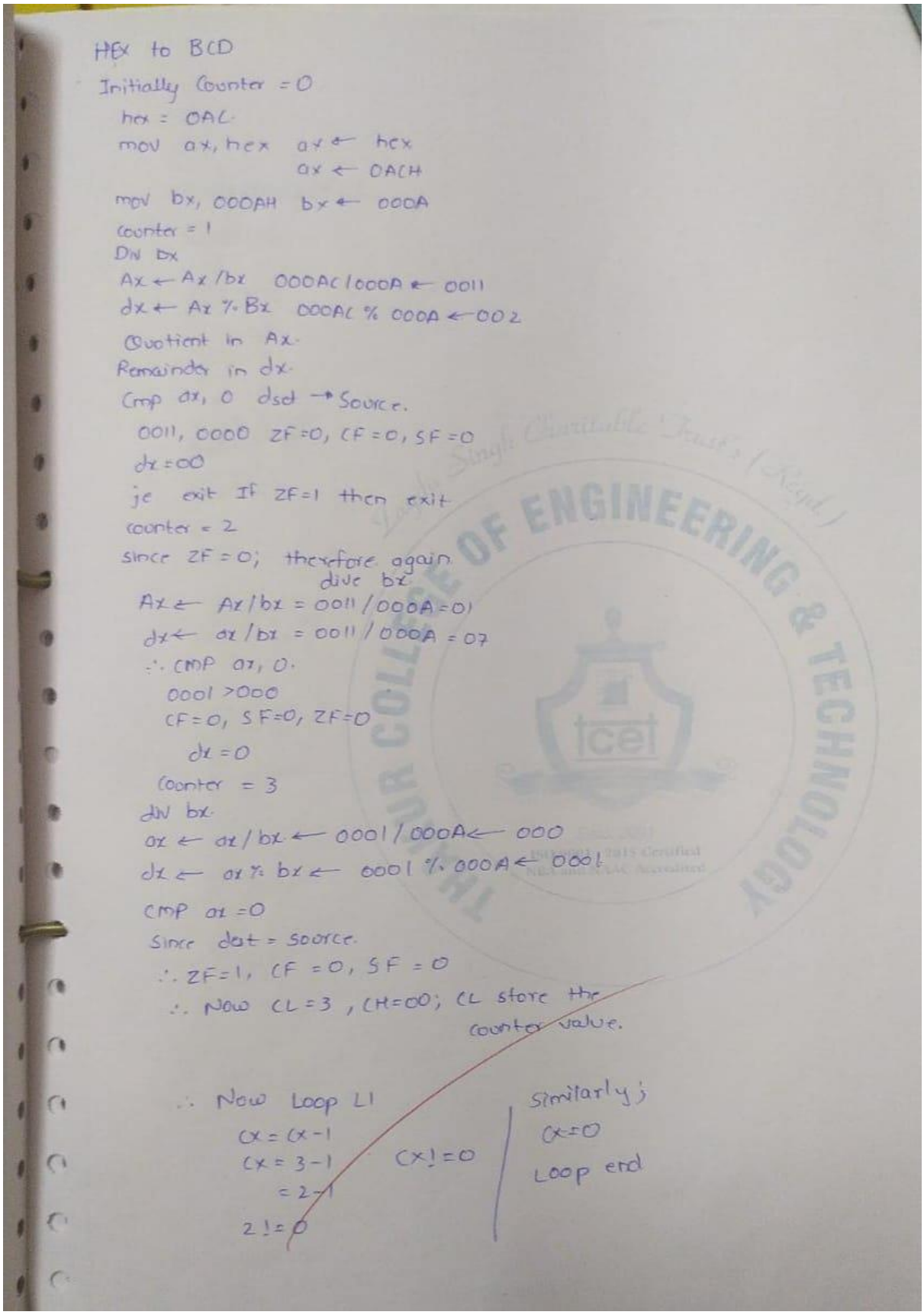
BH → 03H
ADD 30
+03
33H → DL

Addition/Subtraction





Hex to bcd



HEX to BCD

Initially Counter = 0

hex = 0AC

mov ax, hex ax ← hex
ax ← 0ACH

mov bx, 000AH bx ← 000A

counter = 1

DIV bx

Ax ← Ax / bx 000AC / 000A ← 0011

dx ← Ax % Bx 000AC % 000A ← 002

Quotient in Ax.

Remainder in dx.

Cmp ax, 0 dst → Source.

0011, 0000 ZF=0, CF=0, SF=0

dx=00

je exit If ZF=1 then exit

counter = 2

since ZF=0; therefore again
divide bx

Ax ← Ax / bx = 0011 / 000A = 01

dx ← ax / bx = 0011 / 000A = 07

∴ CMP ax, 0.

0001 > 000

CF=0, SF=0, ZF=0

dx = 0

counter = 3

DIV bx.

ax ← ax / bx ← 0001 / 000A ← 000

dx ← ax % bx ← 0001 % 000A ← 0001

CMP ax = 0

Since dst = source.

∴ ZF=1, CF=0, SF=0

∴ Now CL=3, CH=00; CL store the counter value.

∴ Now Loop L1

cx = cx - 1

cx = 3 - 1

= 2 - 1

2 != 0

cx != 0

Similarly;

cx=0

Loop end

Negative No

How to declare Uninitilized Array:

Syntax:

(Example) a DB 7 DUP(0)

where: 'a' → Name of array given.

DB → It is written on the space of datatype which represents which type of data we are considering eg DB → 8 bits DW → 16 bits of data

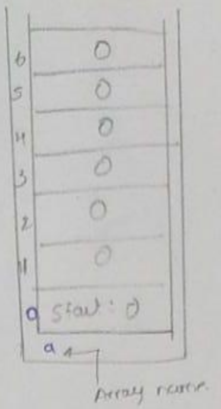
7 → It represents Size of the array assign.

dub(0) → It used for uninitilize array declaration It means assign 0 to the space mentioned for array.

Syntax:

array_name Data_type Size_of_array DUP(0)

* How Array is created for unitilized declaration [a DB 7 DUP(0)]



* How to Declare Initialized Array:

Syntax:

Array_name Data_type Array_Elements

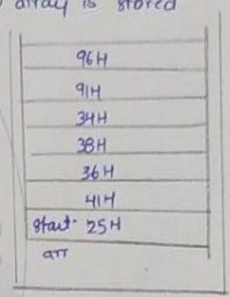
Array_name: Name assigned to array.

Data_type: The data type of the array elements.

Array_Elements: The value of the array which is to be stored in Array_name

Example: arr DB 25H, 41H, 36H, 38H, 34H, 91H, 96H

* How array is stored



* To Check if a number is negative

Method 1: Shift Left by 1

If after performing Left shift operation the carry flag (CF) is set to 1 the number is negative, else if CF=0 then number is positive.

	Cy			Cy		
Eg1	0	10010101	Shl by 1	1	00101010	⇒ Negative
Eg2	0	01100110	Shl by 1	0	11001100	⇒ positive.

Method 2: AND with 80H

when we perform AND operation with 80H to any number and if zeroflag is set to 1 (ZF=1) then number is positive else for ZF=0 the number is negative

Eg

```
  10101100
  10000000
  --------
  1000 0000
```

ZF=0

∴ Negative number.

```
  0011 1111
  1000 0000
  ---------
  0000 0000
```

ZF = 1

∴ Positive Number.

String operation

1. How to accept string from user:

function: MOV AH, 0AH
LEA DX, Str-variable
INT 21H

2) How to declare string in data structure.

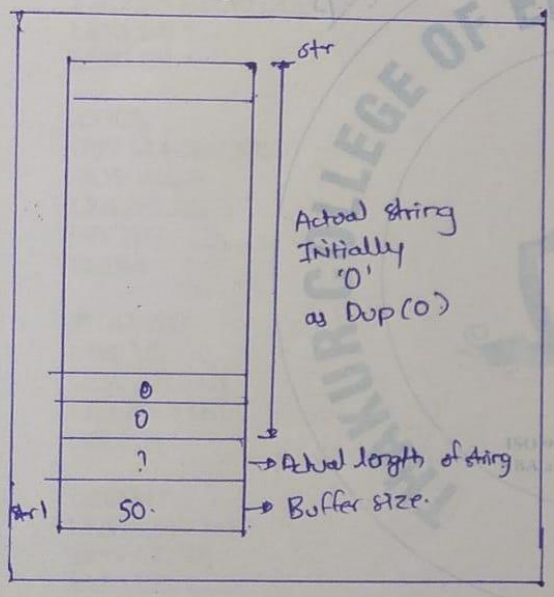
format / syntax

Name-str Data Type Buffer. ?, Actualsize Dup(0)

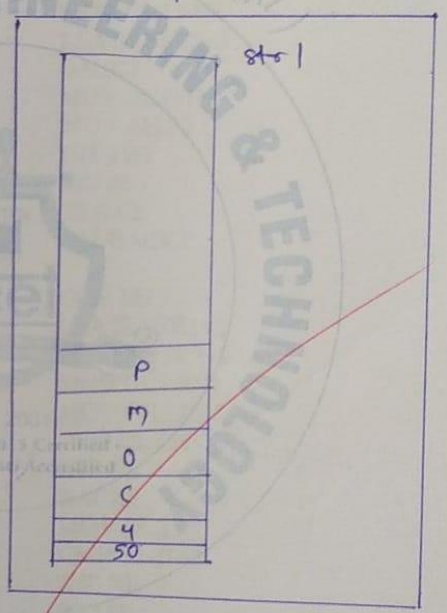
Ex:- str1 DB 50, ?, 50 Dup(0)

Result and Discussion:

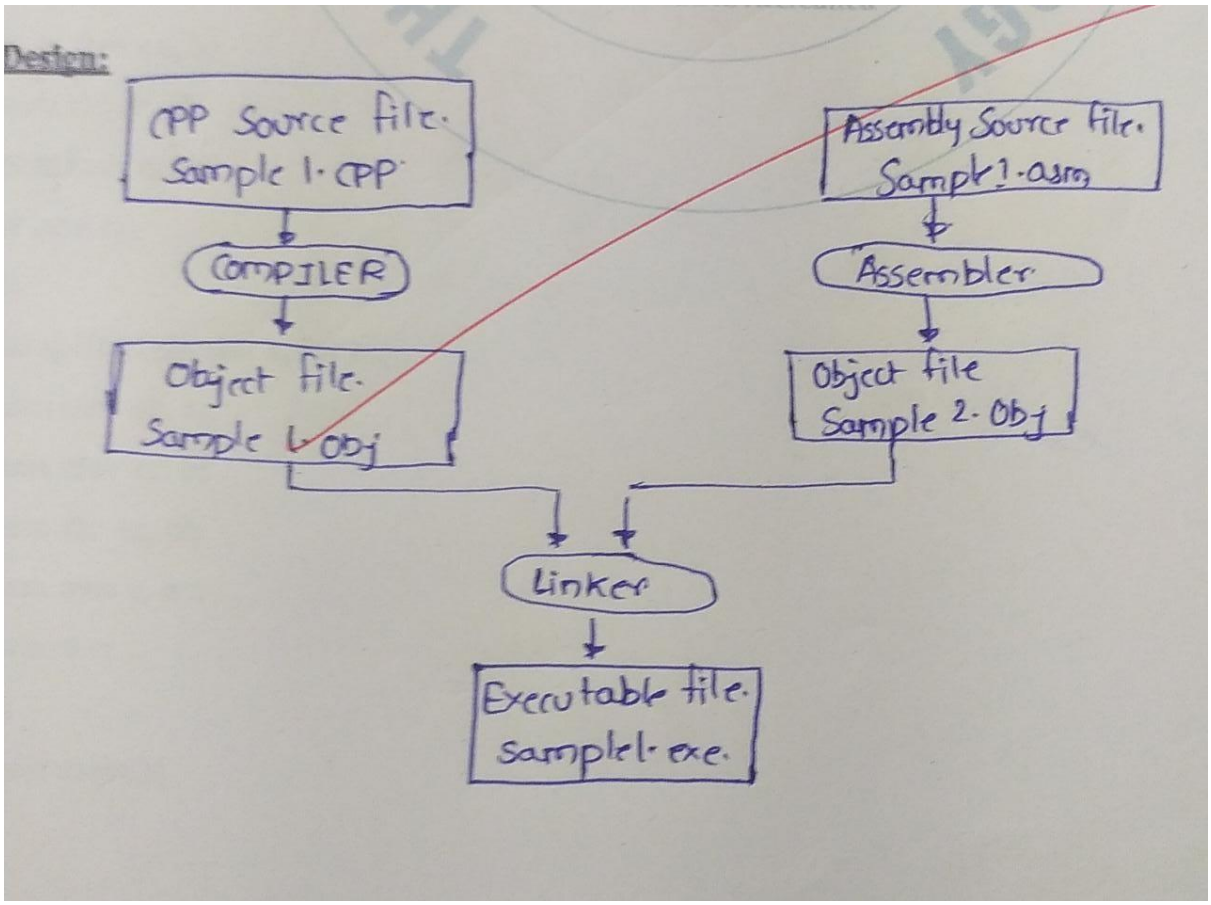
Data structure:



Input : comp



Mix mode


Design:
CPP Source file.
Sample 1.CPP
COMPILER
Object file.
Sample 1.obj
Assembly Source file.
Sample1.asm
Assembler
Object file
Sample 2.Obj
Linker
Executable file.
sample1.exe.

Mouse

Design

Interrupt to access the mouse driver

INT 33hex

1. AX = 0000 → checks status of INT 33H mouse driver
   Returns for.
   AX ≠ 0; mouse driver present
   AX = 0; mouse driver not present.

2. AX = 0001; To check mouse.
   INT 33H Cursor/Show the mouse cursor

3. AX = 0004H; To initialize the mouse cursor position
   ; This function will store
   ; Cx = X - co-ordinate
   ; Dx = Y - co-ordinate
   INT 33H.

4. AX = 0007H
   ; It is used to set horizontal limit
   ; Cx = min x-co-ordinate
   ; Dx = max x-coordinate
   INT 33H.

5. AX = 0008H
   ; It is used to set the vertical limit
   ; Cx = min Y-co-ordinate
   ; Dx = max Y-co-ordinate
   INT 33H

6. AX = 0003H
   ; used to get mouse button status
   INT 33H

Returns →
Bx = 1 (left button clicked)
Bx = 0 (Center button clicked)
Bx = 2 (Right button Clicked)

7) Text mode (To set Text mode)
   AX = 0000H
   INT 10H

8) To set graphics mode.
   Ax = 0011H
   INT 10H

9) To display pixel
   AH = 0CH
   INT 10H.