Digital Design Verification

Lab # 02

Decision Making, Conditionals

Submitted by:

Name:	**Khalil Rehman**
Instructor:	**Hira Sohail**

Date:
July 15, 2025

NUST Chip Design Centre (NCDC), Islamabad, Pakistan

TASK # 01:

FLOW CHART:
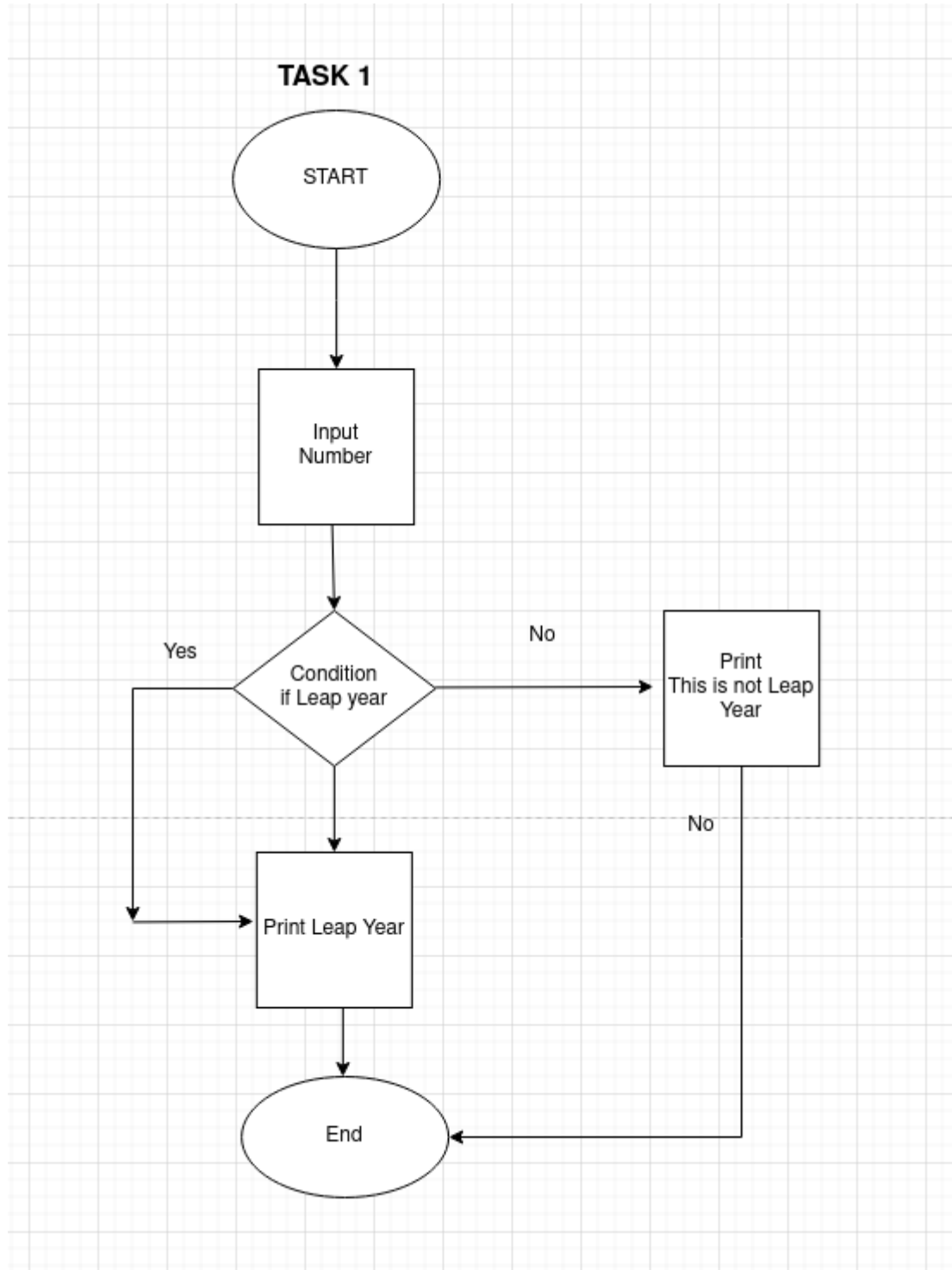
TASK 1

START

Input Number

Condition if Leap year

Yes → Print Leap Year

No → Print This is not Leap Year

Print Leap Year

End

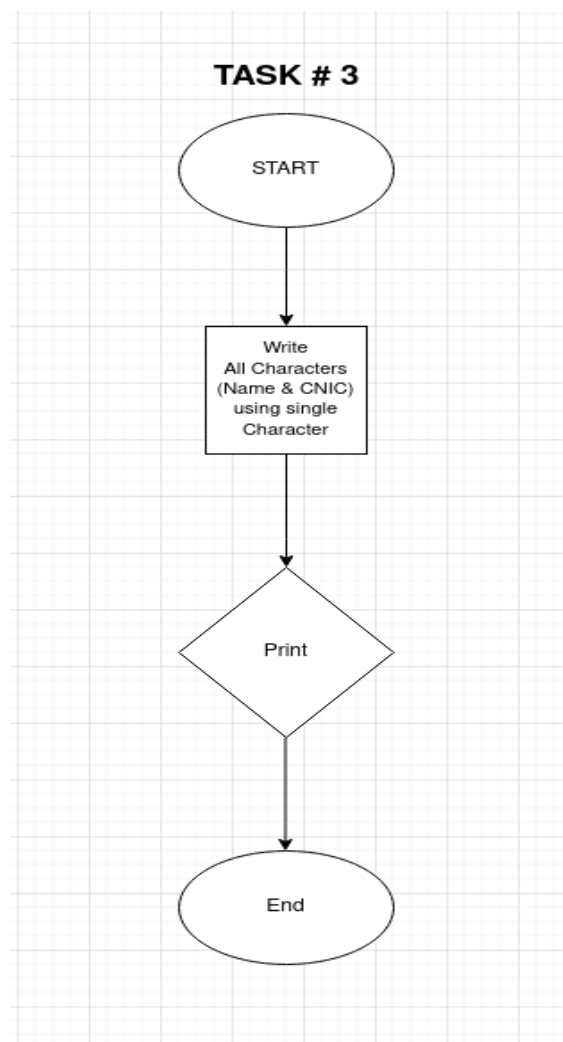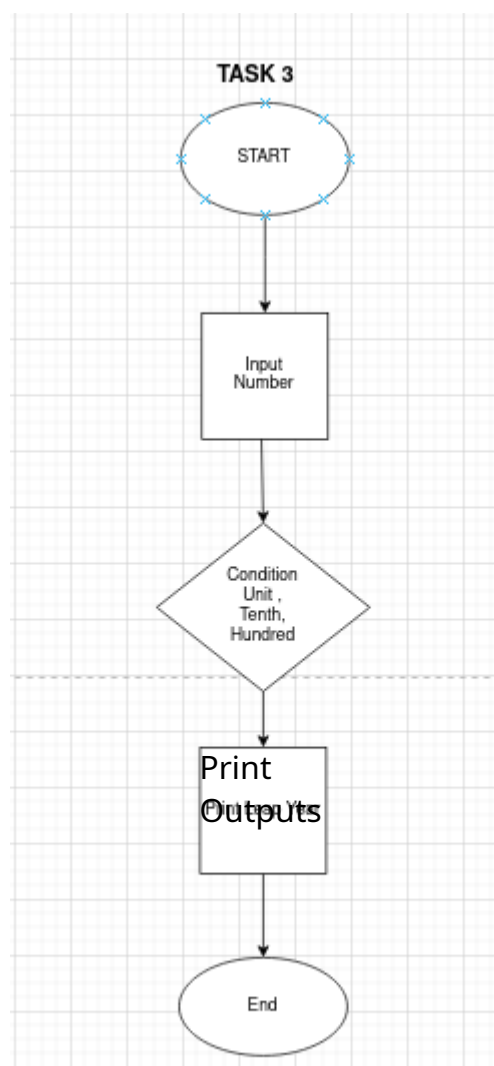**Terminal output:**

TASK # 02:

Flow Chart:

TERMINAL OUTPUT:


```
cc@ncdc-0087:~/Documents/KR/lab2

File  Edit  View  Search  Terminal  Help

[cc@ncdc-0087 lab2]$ gcc task2.c -o task2
[cc@ncdc-0087 lab2]$ ./task2
Enter Degree:35
Enter Minutes:400
Enter Seconds:45

Latitute & Longitute in Decimal Degree: 41.000000
It is above the equator:
[cc@ncdc-0087 lab2]$
```

TASK # 03:

Flow Chart:
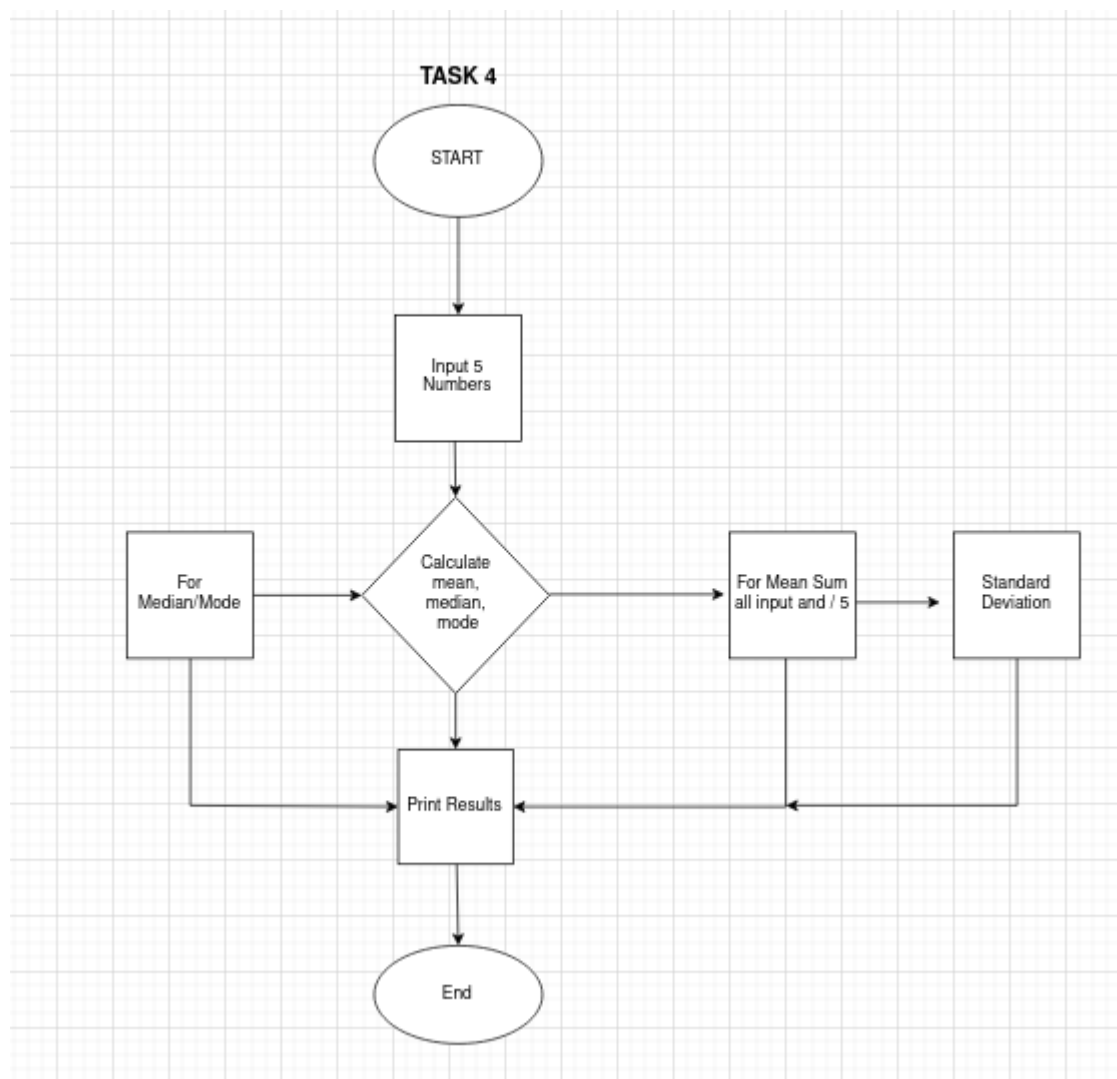
Terminal output:


```
cc@ncdc-0087:~/Documents/KR/lab2                    ×

File  Edit  View  Search  Terminal  Help
[cc@ncdc-0087 lab2]$ ./task3
Enter number:5555

Unit Digit:5.000000
Tenth Digit:5.000000
Hundred Digit:55.000000
[cc@ncdc-0087 lab2]$ 
```

TASK # 04:

Flow Chart:

TERMINAL OUTPUT:
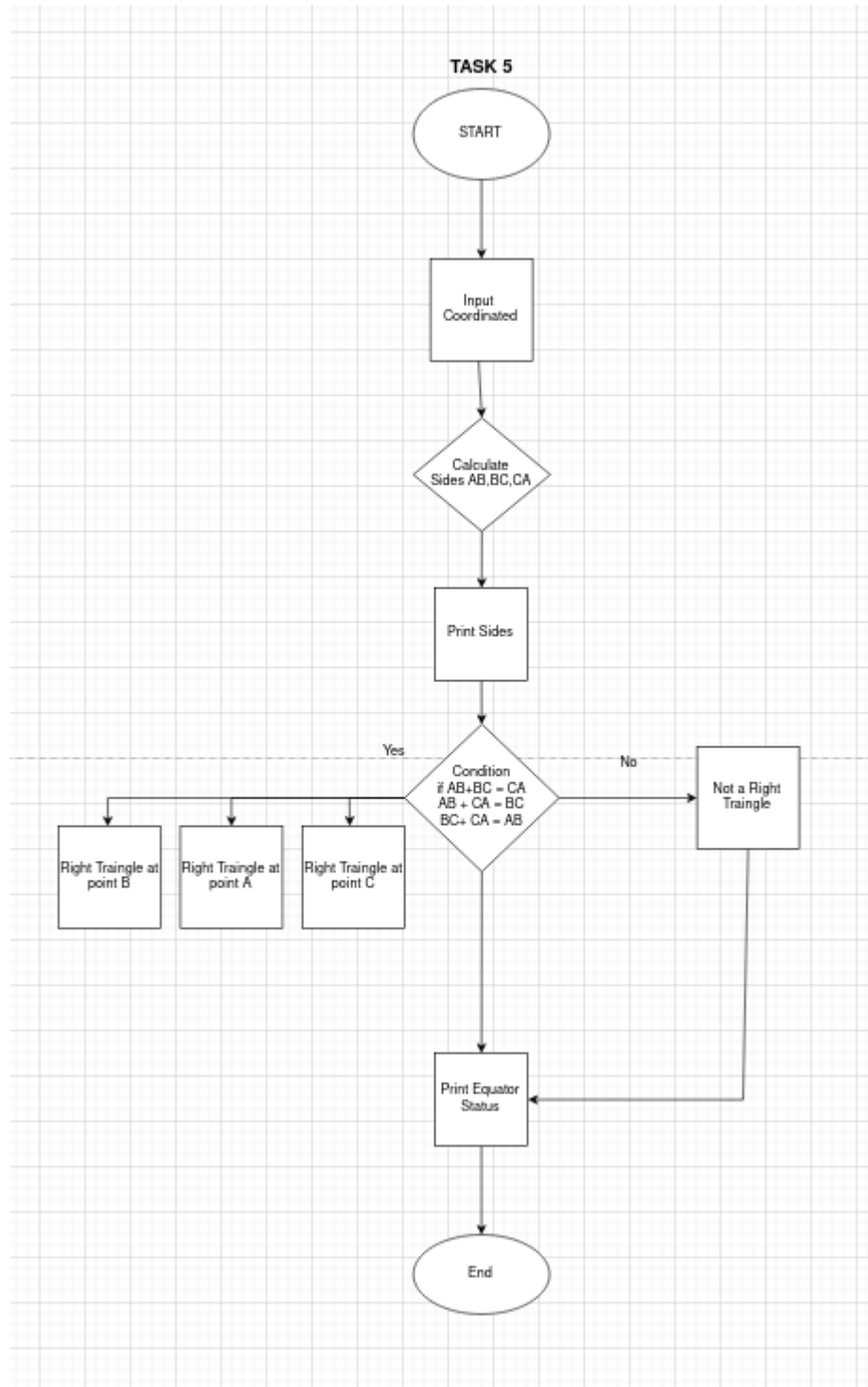


```
cc@ncdc-0087:~/Documents/KR/lab2                              ×

File  Edit  View  Search  Terminal  Help
[cc@ncdc-0087 lab2]$ gcc task4.c -o task4 -lm
[cc@ncdc-0087 lab2]$ ./task4
Enter 5 numbers:
1
2
3
4
5

Mean = 3.000000
Median = 3.000000
Mode = No mode (all unique)
Standard Deviation = 1.41
[cc@ncdc-0087 lab2]$
```

TASK # 05:

FLOW CHART:

TERMINAL OUTPUT:


```
cc@ncdc-0087:~/Documents/KR/lab2                              ×

File  Edit  View  Search  Terminal  Help
[cc@ncdc-0087 lab2]$ gcc task5.c -o task5
[cc@ncdc-0087 lab2]$ ./task5
Enter X1:0
Enter X2:4
Enter X3:0
Enter Y1:0
Enter Y2:0
Enter Y3:3

Enter AB:16
Enter BC:25
Enter CA:9
Right triangle at point A
[cc@ncdc-0087 lab2]$
```
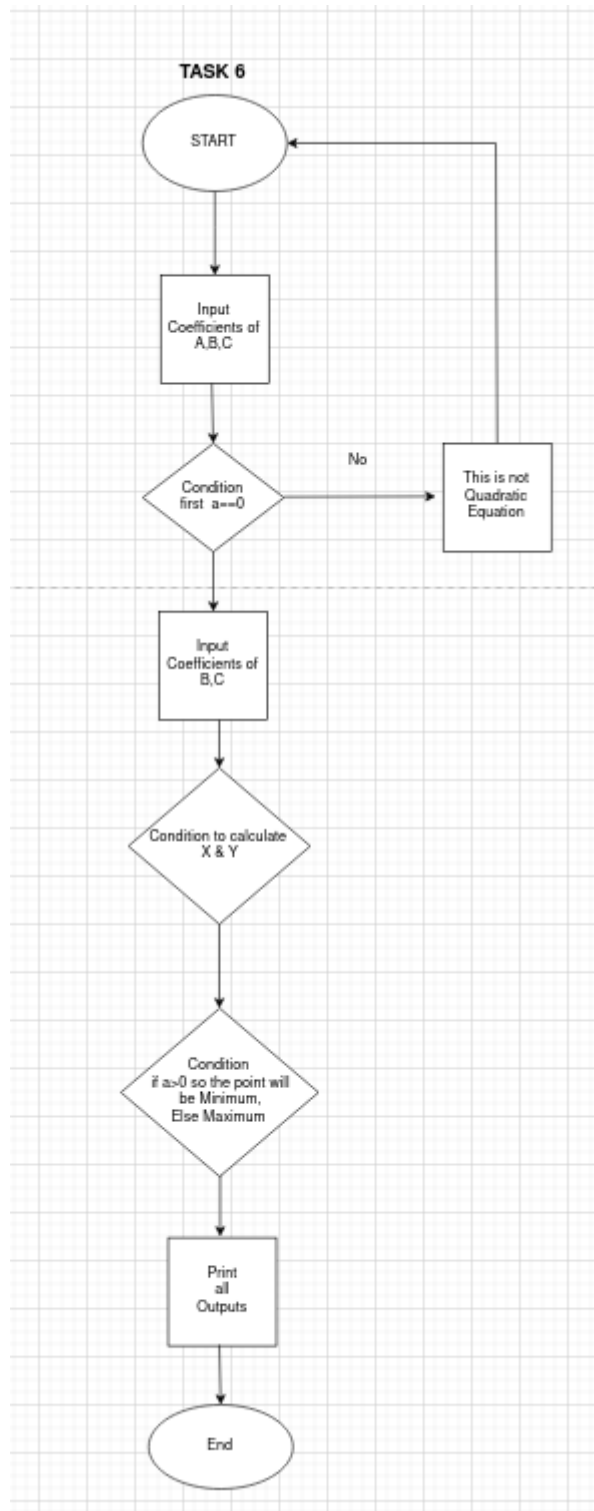
TASK # 06

FLOW CHART:

TERMINAL OUTPUT:

TASK # 07

TERMINAL OUTPUT:

Using C online Compiler

Output

```
Testing get_bit()

get_bit(0x0000004e,0)returned 0x00000000, correct
get_bit(0x0000004e,1)returned 0x00000001, correct
get_bit(0x0000004e,5)returned 0x00000000, correct
get_bit(0x0000001b,3)returned 0x00000001, correct
get_bit(0x0000001b,2)returned 0x00000000, correct
get_bit(0x0000001b,9)returned 0x00000000, correct


Testing set_bit()

set_bit(0x0000004e,2,0) returned 0x0000004a, correct
set_bit(0x0000006d,0,0) returned 0x0000006c, correct
set_bit(0x0000004e,2,1) returned 0x0000004e, correct
set_bit(0x0000006d,0,1) returned 0x0000006d, correct
set_bit(0x0000004e,9,0) returned 0x0000004e, correct
set_bit(0x0000006d,4,0) returned 0x0000006d, correct
set_bit(0x0000004e,9,1) returned 0x0000024e, correct
set_bit(0x0000006d,7,1) returned 0x000000ed, correct


Testing flip_bit()

flip_bit(0x0000004e,0) returned 0x0000004f, correct
flip_bit(0x0000004e,1) returned 0x0000004c, correct
flip_bit(0x0000004e,2) returned 0x0000004a, correct
flip_bit(0x0000004e,5) returned 0x0000006e, correct
flip_bit(0x0000004e,9) returned 0x0000024e, correct



=== Code Execution Successful ===
```