RN SHETTY TRUST®
RNS INSTITUTE OF TECHNOLOGY

Autonomous Institution, Affiliated to VTU, Recognized by GOK, Approved by AICTE
(NAAC 'A+ Grade' Accredited, NBA Accredited (UG - CSE, ECE, ISE, EIE and EEE)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098
Ph:(080)28611880,28611881 URL: www.rnsit.ac.in

DEPARTMENT OF ISE

SOFTWARE TESTING LAB MANUAL

(21ISL66)

Compiled by

DEPARTMENT OF ISE
R N S Institute of Technology
Bengaluru-98

RN SHETTY TRUST®

RNS INSTITUTE OF TECHNOLOGY

Autonomous Institution, Affiliated to VTU, Recognized by GOK, Approved by AICTE
(NAAC 'A+ Grade' Accredited, NBA Accredited (UG - CSE, ECE, ISE, EIE and EEE)
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098
Ph:(080)28611880,28611881 URL: www.rnsit.ac.in

DEPARTMENT OF ISE

VISION OF THE DEPARTMENT

Building Information Technology Professionals by Imparting Quality Education and Inculcating Key Competencies

MISSION OF THE DEPARTMENT

- Provide strong fundamentals through learner centric approach
- Instil technical, interpersonal, interdisciplinary skills and logical thinking for holistic development
- Train to excel in higher education, research, and innovation with global perspective
- Develop leadership and entrepreneurship qualities with societal responsibilities

Disclaimer

The information contained in this document is the proprietary and exclusive property of RNS Institute except as otherwise indicated. No part of this document, in whole or in part, may be reproduced, stored, transmitted, or used for course material development purposes without the prior written permission of RNS Institute of Technology.

The information contained in this document is subject to change without notice. The information in this document is provided for informational purposes only.

Trademark

Estd : 2001

Edition: 2023- 24

Document Owner

The primary contact for questions regarding this document is:

Author(s):

1. Dr. Prakasha S
2. Prof. Ravi kumar S G
3. Prof. Vanishree S

COURSE OUTCOMES

Course Outcomes: At the end of this course, students are able to:
CO1- List out the requirements for the given problem and develop test cases for any given problem .
CO2- Design and implement the solution for given problem and to design flow graph
CO3- Use Eclipse/NetBeans IDE and testing tools to design, develop, debug the Project and create appropriate document for the software artifact.
CO4- Use the appropriate functional testing strategies. Compare the different testing techniques.
CO5-Classify and Compare the problems according to a suitable testing model applying the test coverage metrics.

COs and POs Mapping of lab Component

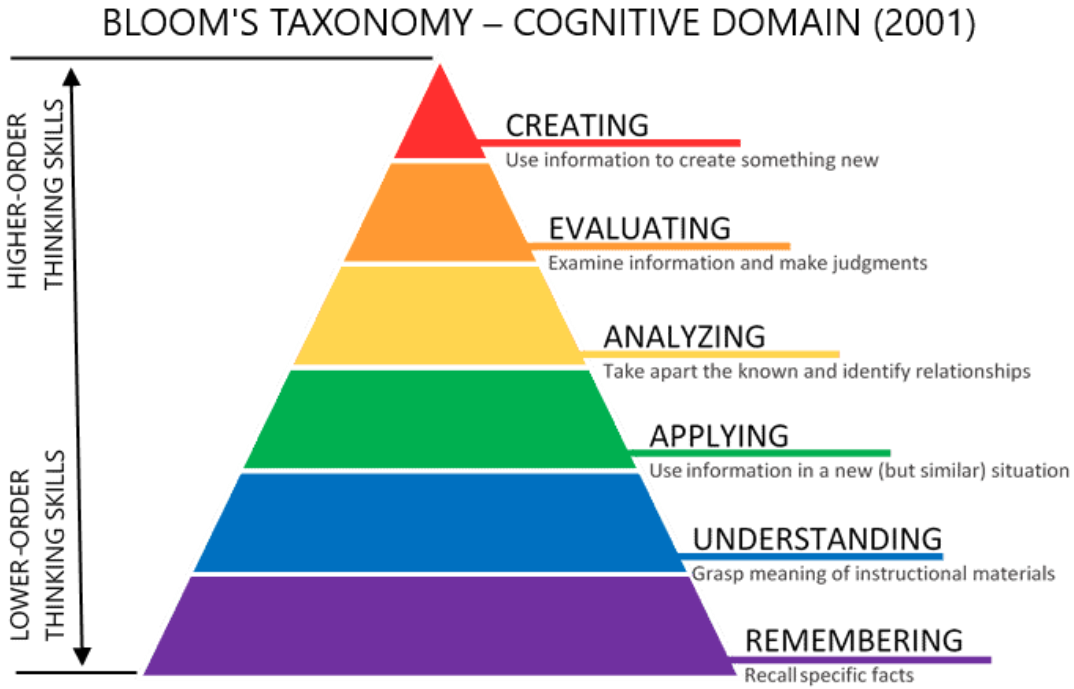
COURSE OUTCOMES	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	3	3	2	2	2				2				
CO2	3	3	3	3	3	2	2	2				2				
CO3	3	3	3	3	3	2	2	2				2				
CO4	3	3	3	3	3	2	2	2				2				

Mapping of 'Graduate Attributes' (GAs) and 'Program Outcomes' (POs)

Graduate Attributes (GAs) (As per Washington Accord Accreditation)	Program Outcomes (POs) (As per NBA New Delhi)
Engineering Knowledge	Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems
Problem Analysis	Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
Design/Development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate considerations for the public health and safety and the cultural, societal and environmental consideration.
Conduct Investigation of complex problems	Use research – based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.
Modern Tool Usage	Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
The engineer and society	Apply reasoning informed by the contextual knowledge to assess society, health, safety, legal and cultural issues and the consequential responsibilities relevant to the professional engineering practice.

Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental context and demonstrate the knowledge of and need for sustainable development.
Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
Individual and team work	Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.
Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.
Project management & finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to ones won work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
Life Long Learning	Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

REVISED BLOOMS TAXONOMY (RBT)


BLOOM'S TAXONOMY – COGNITIVE DOMAIN (2001)
HIGHER-ORDER THINKING SKILLS
LOWER-ORDER THINKING SKILLS
CREATING
Use information to create something new
EVALUATING
Examine information and make judgments
ANALYZING
Take apart the known and identify relationships
APPLYING
Use information in a new (but similar) situation
UNDERSTANDING
Grasp meaning of instructional materials
REMEMBERING
Recall specific facts

PROGRAM LIST

6	Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.	***16***
7	**PART B –** **Practical Based Learning**	***19***
1	Develop a Mini Project with documentation of suitable test-cases and their results to perform automation testing of any E-commerce or social media web page.	***23***

1. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.

/* Assumption price for lock=45.0, stock=30.0 and barrels=25.0, production limit that could be sold in a month is 70 locks, 80 stocks and 90 barrels. Commission on sales = 10 % on sales <= 1000 and 15 % on 1001 to 1800 and 20 % on above 1800*/

```
#include<stdio.h>
int main()
{
        Int locks, stocks, barrels, tlocks, tstocks, tbarrels;
        float lprice, sprice, bprice, sales, comm;
      int c1,c2,c3,temp;
        lprice=45.0;
        sprice=30.0;
        bprice=25.0;
        tlocks=0;
        tstocks=0;
```

```
tbarrels=0;
printf("\n enter the number of locks and to exit the loop enter -1 for locks\n");
scanf("%d", &locks);
while (locks! = -1)
                                                                                    {
        c1= (locks<=0 || locks>70);
        printf("enter the number of stocks and barrels\n");
        scanf("%d%d", &stocks, &barrels);
        c2=(stocks<=0 || stocks>80);
        c3=(barrels<=0 || barrels>90);
        if(c1)




        printf("value of locks not in the range 1..70  ");
                else
        {
                temp=tlocks+locks;
                if(temp>70)
                                printf("new total locks =%d not in the range 1..70 ", temp);
                else
    tlocks=temp;
        }
        printf("total locks = %d\n", tlocks);

  if(c2)
                printf("value of stocks not in the range 1..80  ");
```

```
                else
                {
                        temp=tstocks+stocks;
                        if(temp>80)
                                        printf("new total stocks =%d  not in the range 1..80 ", temp);
                                else
                                tstocks=temp;
        }
                printf("total stocks=%d\n", tstocks);

                if(c3)
                        printf("value of barrels not in the range 1..90  ");
            else
                 {
                                temp=tbarrels+barrels;
                                if(temp>90)
                                        printf("new total barrels =%d not in the range 1..90 ", temp);
                                else
                                        tbarrels=temp;
                 }
            printf("total barrels=%d", tbarrels);
            printf("\n enter the number of locks and to exit the loop enter -1 for locks \n");
            scanf("%d", &locks);
    }
    printf("\n total locks = %d\n total stocks =%d\n total barrels =%d\n", tlocks, tstocks, tbarrels);
    sales = lprice*tlocks + sprice*tstocks + bprice*tbarrels;
printf("\n the total sales=%f\n", sales);
    if(sales > 0)
    {
                if(sales > 1800.0)
```

```
                {
                        comm=0.10*1000.0;
                        comm=comm+0.15*800;

                                                                comm=comm+0.20*(sales-1800.0);
                }
                else if(sales > 1000)
                {
                comm =0.10*1000;
                        comm =comm+0.15*(sales-1000.0);
                }
        else
                        comm=0.10*sales;
                                        printf("the commission is=%f\n", comm);
    }
    else
                printf("there is no sales\n");
        return 0;
}
```

Case Id	Description	Input Data			Expected Output		Actual output		Status	Comment
		Total Locks	Total Stocks	Total Barrels	Sales	Comm-ission	Sales	Comm-ission		
1	Set locks and stocks as nominal value and vary barrels value.	35	40	1	2800					
2	Set locks and stocks as nominal value and vary	35	40	2	2825					

	barrels value.									
3	Set locks and stocks as nominal value and vary barrels value.	35	40	45	3900					
4	Set locks and stocks as nominal value and vary barrels value.	35	40	89	5000					
5	Set locks and stocks as nominal value and vary barrels value.	35	40	90	5025					
6	Set locks and barrels as nominal value and vary stocks value	35	1	45	2730					
7	Set locks and barrels as nominal value and vary stocks value	35	2	45	2760					
8	Set locks and barrels as nominal value and vary stocks value	35	40	45	3900					
9	Set locks and barrels as nominal value and vary stocks value	35	79	45	5070					
10	Set locks and barrels as nominal value and vary stocks value	35	80	45	5100					
11	Set stocks and barrels as nominal value and vary locks value	1	40	45	2370					
12	Set stocks and barrels as nominal value and vary locks value	2	40	45	2415					
13	Set stocks and barrels as	35	40	45	3900					

	nominal value and vary locks value									
14	Set stocks and barrels as nominal value and vary locks value	69	40	45	5430					
15	Set stocks and barrels as nominal value and vary locks value	70	40	45	5475					

Commission Problem Output Boundary Value Analysis Test Cases

Case Id	Description	Input Data			Expected Output		Actual output		Status	
		Total Locks	Total Stocks	Total Barrels	Sales	Comm-ission	Sales	Comm-ission		Comment
1	Enter the min value for locks, stocks and barrels	1	1	1	100	10				output minimum
2	Enter the min value for 2 items and min +1 for any one item	1	1	2	125	12.5				output minimum +
3		1	2	1	130	13				output minimum +
4		2	1	1	145	14.5				output minimum +
5	Enter the value sales approximately mid value between 100 to 1000	5	5	5	500	50				Midpoint
6	Enter the values to calculate the commission for sales nearly less than 1000	10	10	9	975	97.5				Border point -
7		10	9	10	970	97				Border point -
8		9	10	10	955	95.5				Border point -
9		10	10	10	1000	100				Border point

	Enter the values sales exactly equal to 1000									
10	Enter the values to calculate the commission for sales nearly greater than 1000	10	10	11	1025	103.75				Border point +
11		10	11	10	1030	104.5				Border point +
12		11	10	10	1045	106.75				Border point +
13	Enter the value sales approximately mid value between 1000 to 1800	14	14	14	1400	160				Midpoint
14	Enter the values to calculate the commission for sales nearly less than 1800	18	18	17	1775	216.25				Border point -
15		18	17	18	1770	215.5				Border point -
16		17	18	18	1755	213.25				Border point -
17	Enter the values sales exactly equal to 1800	18	18	18	1800	220				Border point
18	Enter the values to calculate the commission for sales nearly greater than 1800	18	18	19	1825	225				Border point +
19		18	19	18	1830	226				Border point +
20		19	18	18	1845	229				Border point +
21	Enter the value sales approximately mid value between 1800 to 7800	48	48	48	4800	820				Midpoint
22	Enter the max value for 2 items and max - 1 for any one item	70	80	89	7775	1415				Output maximum -
23		70	79	90	7770	1414				Output maximum -
24		69	80	90	7755	1411				Output maximum -
25	Enter the max value for locks, stocks and barrels	70	80	90	7800	1420				Output maximum

Output Special Value Test Cases

Case Id	Description	Input Data			Expected Output		Actual output			
		Total Locks	Total Stocks	Total Barrels	Sales	Comm-ission	Sales	Com m-ission	Statu s	Comment
1	Enter the random values such that to calculate commission for sales nearly less than 1000	11	10	8	995	99.5				Border point -
2	Enter the random values such that to calculate commission for sales nearly greater than 1000	10	11	9	1005	100.75				Border point +
3	Enter the random values such that to calculate commission for sales nearly less than 1800	18	17	19	1795	219.25				Border point -
4	Enter the random values such that to calculate commission for sales nearly greater than 1800	18	19	17	1805	221				Border point +

2. Design, develop, code and run the program in any suitable language to implement the NextDate function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.

Test Case Name : Equivalence class test cases for NextDate

Experiment Number :6

Test data : Enter the three integer value

Pre-condition : Month 1 to 12 , DAY 1 TO 31 & YEAR 1812 TO 2019

Valid Classes

M1 = { month ; 1 ≤ month ≤ 12 }

D1 = { day : 1 ≤ day ≤ 31 }

Y1 = { year : 1812 ≤ year ≤ 2019 }

Invalid Classes

M2 = {month : month < 1}

M3 = {month : month > 12}

D2 = {day : day < 1}

D3 = {day : day > 31}

Y2 = {year : year < 1812}

Y3 = {year : year > 2019}

NextDate Equivalence Class Testing

(Weak and Strong Normal Equivalence Class)

Case Id	Description	Input Data	Expected Output	Actual output	Status	Comment

		month	day	year	month	day	year	month	day	year		
WN1, SN1	Enter the valid value for month, day and year	6	15	1915	6	16	1915					

(Weak Robust Equivalence Class)

Case Id	Description	Input Data			Expected Output			Actual output			Status	Comment
		month	day	year	month	day	year	month	day	year		
WR1	Enter the valid value for month, day and year	6	15	1915	6	16	1915					
WR2	Enter the invalid value for month and valid value for day and year	-1	15	1915	Should display the message value of the month not in the range 1..12							
WR3	Enter the invalid value for month and valid value for day and year	13	15	1915	Should display the message value of the month not in the range 1..12							
WR4	Enter the invalid value for day and valid value for month and year	6	-1	1915	Should display the message value of the day not in the range 1..31							
WR5	Enter the invalid value for day and valid value for month and year	6	32	1915	Should display the message value of the day not in the range 1..31							

WR6	Enter the invalid value for year and valid value for month and day	6	15	1811	Should display the message value of the year not in the range 1812..2017			
WR7	Enter the invalid value for year and valid value for month and day	6	15	2020	Should display the message value of the year not in the range 1812..2019			

(Strong Robust Equivalence Class)

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comment
		month	day	year				
SR1	Enter the invalid value for month and valid value for day and year	-1	15	1915	Should display the message value of the month not in the range 1..12			
SR2	Enter the invalid value for day and valid value for month and year	6	-1	1915	Should display the message value of the day not in the range 1..31			
SR3	Enter the invalid value for year and valid value for month and day	6	15	1811	Should display the message value of the year not in the range 1812..2019			
SR4	Enter the invalid value for month and day and valid value for year	-1	-1	1915	(i)Should display the message value of the month not in range 1..12			
					(ii) Should display the message value of the day not in range 1..31			
SR5	Enter the invalid value for day and year and valid value for month	6	-1	1811	**(i)** Should display the message value of the day not in range 1..31			
					(ii) Should display the message value of the year not in range 1812..2019			

SR6	Enter the invalid value for year and month and valid value for day	-1	15	1811	(**i**)Should display the message value of the month not in range 1..12			
					(ii) Should display the message value of the year not in range 1812..2019			
SR7	Enter the invalid value for month, day and year	-1	-1	1811	(**i**)Should display the message value of the month not in range 1..12			
					(ii) Should display the message value of the day not in range 1..31			
					(iii) Should display the message value of the year not in range 1812..2019			

Some addition Equivalence Class Testcases

Case Id	Description	Input Data			Expected Output			Actual Output			Status	Comment
		day	month	year	day	month	year	day	month	year		
1	Enter the invalid value for year valid value for day and month	31	12	1811	Should display the message value of the year not in range 1812..2019							
2	Enter the valid value for month, day and year	31	12	2016	1	1	2017					
3	Enter the valid value for month, day and year	28	2	2000	29	2	2000					

4	Enter the valid value for month, day and year	28	2	1996	29	2	1996					
5	Enter the valid value for month, day and year	29	2	2000	1	3	2000					
6	Enter the valid value for month, day and year	29	2	1996	1	3	1996					
7	Enter the valid value for month, day and year	28	2	2002	1	3	2002					
8	Enter the valid value for month, day and year	29	2	2002	Invalid I/P Date							
9	Enter the invalid value for year, valid value for day and month	31	12	2020								

3. Design, develop, code and run the program in any suitable language to solve the commission problem.

Analyze it from the perspective of decision table-based testing, derive different test cases, execute these test cases and discuss the test results.

Test Case Name :Decision Table for Commission Problem

Experiment Number : 3

Test data : price for lock = 45.0 , stock = 30.0 and barrel = 25.0
sales = total locks * lock price + total stocks * stock price + total barrels * barrel price
commission : 10% up to sales Rs 1000 , 15 % of the next Rs 800 and 20 % on any sales in excess of 1800

Pre-condition : lock = -1 to exit and 1< =lock < = 70 , 1<=stock <=80 and 1<=barrel<=90

Brief Description: The salesperson had to sell at least one complete rifle per month.

Input data decision Table

RULES		R1	R2	R3	R4	R 5	R6	R7	R8	R9
Conditions	C1: Locks = -1	T	F	F	F	F	F	F	F	F
	C2 : 1 ≤ Locks ≤ 70	-	T	T	F	T	F	F	F	T
	C3 : 1 ≤ Stocks ≤ 80	-	T	F	T	F	T	F	F	T
	C4 : 1 ≤ Barrels ≤ 90	-	F	T	T	F	F	T	F	T
Actions	A1 : Terminate the input loop	X								
	A2 : Invalid locks input				X		X	X	X	
	A3 : Invalid stocks input			X		X		X	X	
	A4 : Invalid barrels input		X			X	X		X	
	A5 : Calculate total locks, stocks and barrels		X	X	X	X	X	X		X
	A6: Calculate Sales	X								
	A7: proceed to commission decision table	X								

Commission calculation Decision Table (Precondition : lock = -1)

RULES		R1	R2	R3	R4
Conditions	C1 : Sales = 0	T	F	F	F
	C2 : Sales > 0 AND Sales ≤ 1000		T	F	F
	C3 : Sales > 1000 AND sales ≤ 1800			T	F
	C4 : sales >1800				T

Actions	A1 : Terminate the program	X			
	A2 : comm= 10%*sales		X		
	A3 : comm = 10%*1000 + (sales-1000)*15%			X	
	A4 : comm = 10%*1000 + 15% * 800 + (sales-1800)*20%				X

Experiment Number : 3

Test data : price for lock = 45.0 , stock = 30.0 and barrel = 25.0
sales = total locks * lock price + total stocks * stock price + total barrels * barrel price
commission : 10% up to sales Rs 1000 , 15 % of the next Rs 800 and 20 % on any sales in excess of 1800
Pre-condition : lock = -1 to exit and 1< =lock < = 70 , 1<=stock <=80 and 1<=barrel<=90
Brief Description: The salesperson had to sell at least one complete rifle per month.

Precondition : Initial Value Total Locks= 0 , Total Stocks=0 and Total Barrels=0

Precondition Limit :Total locks, stocks and barrels should not exceed the limit 70,80 and 90 respectively

Commission Problem -Decision Table Test cases for input data

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comments
		Locks	Stocks	Barrels				
1	Enter the value of Locks= -1	-1			Terminate the input loop check for sales if(sales=0) exit from program else calculate commission			
2	Enter the valid input for locks and stocks and invalid for barrels	20	30	-5	Total of locks, stocks is updated if it is within a precondition limit and Should display value of barrels is not in the range 1..90			
3	Enter the valid input for locks and barrrels and invalid for stocks	15	-2	45	Total of locks, barrels is updated if it is within a precondition limit and Should display value of stocks is not in the range 1..80			
4	Enter the valid input for stocks and barrrels and invalid for locks	-4	15	16	Total of stocks , barrels is updated if it is within a precondition limit and Should display value of locks is not in the range 1..70			

5	Enter the valid input for locks and invalid value for stocks and barrels	15	81	100	Total of locks is updated if it is within a precondition limit and **(i)**Should display value of stock is not in the range 1..80 **(ii)**Should display value of barrels is not in the range 1..90			
6	Enter the valid input for stocks and invalid value for locks and barrels	88	20	99	Total of stocks is updated if it is within a precondition limit and **(i)**Should display value of lock is not in the range 1..70 **(ii)**Should display value of barrels is not in the range 1..90			
7	Enter the valid input for barrels and invalid value for locks and stocks	100	200	25	Total of barrels is updated if it is within a precondition limit and **(i)**Should display value of lock is not in the range 1..70 **(ii)**Should display value of stocks is not in the range 1..80			
8	Enter the invalid input for lock , stocks and barrels	-5	400	-9	**(i)**Should display value of lock is not in the range 1..70 **(ii)**Should display value of stocks is not in the range 1..80 **(iii)**Should display value of barrel in not in the range 1..90			
9	Enter the valid input for lock, stocks and barrels	15	20	25	Total of locks,stocks and barrels is updated if it is within a precondition limit and calculate the sales and proceed to commission			

Commission Problem -Decision Table Test cases for commission calculation

Precondition : Locks = -1

Case Id	Description	Input Data	Expected Output		Actual Output	Status	Comments
		Sales	Commission	Values			
1	Check the value of sales	0	Terminate the program where commission is zero	0			
2	if sales value within these range(Sales >0 AND Sales ≤ 1000)	900	Then commission = 0.10*sales	90			

3	if sales value within these range(Sales > 1000 AND Sales ≤ 1800)	1400	Then commission = 0.10*1000 + 0.15*(sales - 1000)	160			
4	if sales value within these range(Sales > 1800	2500	Then commission = 0.10*1000 + 0.15*800 + 0.20 *(sales - 1800)	340			

4. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis, equivalence class partitioning and decision-table approach and execute the test cases and discuss the results

```
#include<stdio.h>
int main()
{
        int a,b,c,c1,c2,c3;
        char istriangle;
        do
        {
                printf("\n enter 3 integers which are sides of triangle\n");
                scanf("%d%d%d", &a, &b, &c);
                  printf("\n a=%d\t b=%d\t c=%d", a, b, c);
                c1=a>=1 &&  a<=10;
                c2= b>=1 &&  b<=10;
                c3= c>=1 &&  c<=10;
                 if (!c1)
```

```
                printf("\n the value of a=%d is not the range of permitted value", a);
            if (!c2)
                printf("\n the value of b=%d is not the range of permitted value", b);
            if (!c3)
                printf("\n the value of c=%d is not the range of permitted value", c);
    } while(!(c1 && c2 && c3));

    // to check is it a triangle or not

    if( a<b+c && b<a+c && c<a+b )
        istriangle='y';
    else
        istriangle ='n';
    if (istriangle=='y')
        if ((a==b) && (b==c))
            printf("equilateral triangle\n");
        else if ((a!=b) && (a!=c) && (b!=c))
            printf("scalene triangle\n");
          else
            printf("isosceles triangle\n");
    else
        printf("Not a triangle\n");
    return 0;
}
```

Test Case Name :Boundary Value Analysis for triangle problem

Experiment Number : 4

Test Data : Enter the 3 Integer Value(a , b And c)

Pre-condition : $1 \le a \le 10$, $1 \le b \le 10$ and $1 \le c \le 10$ and $a < b + c$, $b < a + c$ and $c < a + b$

Brief Description : Check whether given value for a Equilateral, Isosceles , Scalene triangle or can't form a triangle

Triangle Problem -Boundary value Test cases for input data

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comments
		a	b	c				
1	Keep a and b at nominal value and vary c	5	5	1	Should display the message Isosceles triangle			
2	Keep a and b at nominal value and vary c	5	5	2	Should display the message Isosceles triangle			
3	Keep a and b at nominal value and vary c	5	5	5	Should display the message Equilateral triangle			
4	Keep a and b at nominal value and vary c	5	5	9	Should display the message Isosceles triangle			
5	Keep a and b at nominal value and vary c	5	5	10	Should display the message Not a triangle			
6	Keep a and cat nominal value and vary b	5	1	5	Should display the message Isosceles triangle			
7	Keep a and c at nominal value and vary b	5	2	5	Should display the message Isosceles triangle			
8	Keep a and c at nominal value and vary b	5	5	5	Should display the message Equilateral triangle			
9	Keep a and c at nominal value and vary b	5	9	5	Should display the message Isosceles triangle			
10	Keep a and c at nominal value and vary b	5	10	5	Should display the message Not a triangle			
11	Keep b and cat nominal value and vary a	1	5	5	Should display the message Isosceles triangle			

12	Keep b and c at nominal value and vary a	2	5	5	Should display the message Isosceles triangle			
13	Keep b and c at nominal value and vary a	5	5	5	Should display the message Equilateral triangle			
14	Keep b and c at nominal value and vary a	9	5	5	Should display the message Isosceles triangle			
15	Keep b and c at nominal value and vary a	10	5	5	Should display the message Not a triangle			

Triangle Problem Worst-Case-Test Cases (one corner of a triangle)

Case	Description	a	b	c	Expected Output	Actual Output	Status	Comments
1	Enter the **min value** for a , b and c	**1**	1	**1**	Should display the message as Equilateral triangle			
2	Enter the **min value** for 2 items and **min +1** for any one item	**1**	1	**2**	Should display the message as Not a Triangle			
3	Enter the **min value** for 2 items and **Average value** for any one item	**1**	1	**5**	Should display the message as Not a Triangle			
4	Enter the **min value** for 2 items and **Max -1** for any one item	**1**	1	**9**	Should display the message as Not a Triangle			
5	Enter the **min value** for 2 items and **Max** for any one item	**1**	1	**10**	Should display the message as Not a Triangle			
6	Enter the **min value** for 2 items and **min +1** for any one item	**1**	2	**1**	Should display the message as Not a Triangle			
7	Enter the **min+1 value** for 2 items and **min** for any one item	**1**	2	**2**	Should display the message as Isosceles			
8	Enter the **min value** for **1 items, min+1** and **Average value** for any one item	**1**	2	**5**	Should display the message as Not a Triangle			

9	Enter the **min value** for **1 items, min+1** and **max-1** for any one item	**1**	2	**9**	Should display the message as Not a Triangle			
10	Enter the **min value** for **1 items, min+1** and **max** for any one item	**1**	2	**10**	Should display the message as Not a Triangle			
11	Enter the **min value** for 2 **items, average value** for any one item	**1**	5	**1**	Should display the message as Not a Triangle			
12	Enter the **min value** for **1 items, min+1** and **average** for any one item	**1**	5	**2**	Should display the message as Not a Triangle			
13	Enter the **min value** for **1 items,** and **average** for any 2 items	**1**	5	**5**	Should display the message as Isosceles			
14	Enter the **min value** for **1 items, max-1** and **average** for any one item	**1**	5	**9**	Should display the message as Not a Triangle			
15	Enter the min value for **1 items, max** and **average** for any one item	**1**	5	**10**	Should display the message as Not a Triangle			
16	Enter the **min value** for 2 items and **max -1** for any one item1	**1**	9	**1**	Should display the message as Not a Triangle			
17	Enter the **min value** for **1 items, min+1** and **max-1** for any one item	**1**	9	**2**	Should display the message as Not a Triangle			
18	Enter the **min value** for **1 items, max-1**and **Average value** for any one item	**1**	9	**5**	Should display the message as Not a Triangle			
19	Enter the **min value** for**1 items, max-1**for 2 items	**1**	9	**9**	Should display the message as Isosceles			
20	Enter the **min value** for **1 items, max-1**and **Max value** for any one item	**1**	9	**10**	Should display the message as Not a Triangle			
21	Enter the **min value** for **2 items** and **max** for any one item	**1**	10	**1**	Should display the message as Not a Triangle			
22	Enter the **min value** for **1 items, min+1** and **max** for any one item	**1**	10	**2**	Should display the message as Not a Triangle			
23	Enter the **min value** for **1 items, max** and **Average value** for any one item	**1**	10	**5**	Should display the message as Not a Triangle			
24	Enter the **min value** for **1 items, max-1** , and **max** for 1 items	**1**	10	**9**	Should display the message as Not a Triangle			
25	Enter the **min value** for **1 items,** and **Max value** for 2 items	**1**	10	**10**	Should display the message as Isosceles			

Special Value Test Cases

Case	Description	a	b	c	Expected Output	Actual Output	Status	Comments
1	Enter the **values** for a , b and c	**5**	8	**6**	Should display the message as Scalene triangle			
2	Enter the **out of boundary value** for a and b and **normal** value for c	**11**	0	**5**	Should display the message as value of a and b not in the permitted range			
3	Enter the **negative value** for a, b and c	**-1**	-4	**-6**	Should display the message as value of a, b and c not in the permitted range			
4	Enter the **values** for a , b and c	**5**	**1**	**10**	Should display the message as Not a Triangle			

Test Case Name :Equivalence Class Analysis for triangle problem

Experiment Number : 4

Test Data: Enter the 3 Integer Value (a, b and c)

Pre-condition : $1 \leq a \leq 10$, $1 \leq b \leq 10$ and $1 \leq c \leq 10$ and $a < b + c$, $b < a + c$ and $c < a + b$

Brief Description : Check whether given value for a Equilateral, Isosceles, Scalene triangle or can't form a triangle

Triangle Problem - Equivalence Class Test cases

Weak and Strong Normal Equivalence class Testing								
Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comments
		a	b	C				
WN1/ SN1	Enter the nom value for a , b and c	5	5	5	Should display the message Equilateral triangle			
WN2/ SN2	Enter the valid value for a , b and c	2	2	3	Should display the message Isosceles triangle			
WN3/ SN3	Enter the valid value for a , b and c	3	4	5	Should display the message Scalene triangle			
WN4/ SN4	Enter the valid value for a , b and c	4	1	2	Message should be displayed can't form a triangle			

Weak Robust Equivalence Class Testing								
WR1	Enter one invalid input and two valid value for a , b and c	-1	5	5	Should display value of a is not in the range of permitted values			
WR2	Enter one invalid input and two valid value for a , b and c	5	-1	5	Should display value of b is not in the range of permitted values			
WR3	Enter one invalid input and two valid value for a , b and c	5	5	-1	Should display value of c is not in the range of permitted values			
WR4	Enter one invalid input and two valid value for a , b and c	11	5	5	Should display value of a is not in the range of permitted values			
WR5	Enter one invalid input and two valid value for a , b and c	5	11	5	Should display value of b is not in the range of permitted values			
WR6	Enter one invalid input and two valid	5	5	11	Should display value of c is not in the range			

	value for a , b and c				of permitted values			
Strong Robust Equivalence class Testing								
SR1	Enter one invalid input and two valid value for a , b and c	-1	5	5	Should display value of a is not in the range of permitted values			
SR2	Enter one invalid input and two valid value for a , b and c	5	-1	5	Should display value of b is not in the range of permitted values			
SR3	Enter one invalid input and two valid value for a , b and c	5	5	-1	Should display value of c is not in the range of permitted values			
SR4	Enter two invalid input and one valid value for a , b and c	-1	-1	5	Should display value of a and b are not in the range of permitted values			
SR5	Enter two invalid input and one valid value for a , b and c	5	-1	-1	Should display value of b and c are not in the range of permitted values			
SR6	Enter two invalid input and one valid value for a , b and c	-1	5	-1	Should display value of a and c are not in the range of permitted values			
SR7	Enter all invalid inputs	-1	-1	-1	Should display value of a, b and c are not in the range of permitted values			

Test Case Name :Decision table for triangle problem

Experiment Number : 4

Test Data : Enter the 3 Integer Value(a , b And c)

Pre-condition : a < b + c , b < a + c and c < a + b

Brief Description : Check whether given value for a equilateral, isosceles , Scalene triangle or can't form a triangle

Input data decision Table

RULES		**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**
Conditions	C1: a < b + c	F	T	T	T	T	T	T	T	T	T	T
	C2 : b < a + c	-	F	T	T	T	T	T	T	T	T	T
	C3 : c < a + b	-	-	F	T	T	T	T	T	T	T	T

	C4 : a = b	-	-	-	T	T	T	T	F	F	F	F
	C5 : a = c	-	-	-	T	T	F	F	T	T	F	F
	C6 : b = c	-	-	-	T	F	T	F	T	F	T	F
Actions	a1 : Not a triangle	X	X	X								
	a2 : Scalene triangle											X
	a3 : Isosceles triangle							X		X	X	
	a4 : Equilateral triangle				X							
	a5 : Impossible					X	X		X			

Triangle Problem -Decision Table Test cases for input data

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comments
		a	b	c				
1	Enter the value of a, b and c Such that a is not less than sum of two sides	20	5	5	Message should be displayed can't form a triangle			
2	Enter the value of a, b and c Such that b is not less than sum of two sides and a is less than sum of other two sides	3	15	11	Message should be displayed can't form a triangle			
3	Enter the value of a, b and c Such that c is not less than sum of two sides and a and b is less than sum of other two sides	4	5	20	Message should be displayed can't form a triangle			
4	Enter the value a, b and c satisfying precondition and a=b, b=c and c=a	5	5	5	Should display the message Equilateral triangle			
5	Enter the value a ,b and c satisfying precondition and a=b and b ≠ c	10	10	9	Should display the message Isosceles triangle			
6	Enter the value a, b and c	5	6	7	Should display the message			

	satisfying precondition and a ≠b , b ≠ c and c ≠ a				Scalene triangle			

5. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.

```
1   //Program 9:(Dataflow Testing  for commission calculation)
2   #include<stdio.h>
3   int main()
4   {
5                               int locks, stocks, barrels, tlocks, tstocks, tbarrels;
6               float lprice, sprice, bprice, lsales, ssales, bsales, sales, comm;
7                       lprice =45.0;
8               sprice=30.0;
9                       bprice=25.0;
10                      tlocks=0;
11                      tstocks=0;
12                      tbarrels=0;
            printf("\nenter the number of locks and to exit the loop enter -1 for locks\n");
13          scanf("%d",&locks);
14          while(locks!= -1){
            printf("enter the number of stocks and barrels\n");
15          scanf("%d%d",&stocks, &barrels);
16                  tlocks = tlocks + locks;
17                              tstocks = tstocks + stocks;
18                  tbarrels = btarrels + barrels;
            printf("\n enter the number of locks and to exit the loop enter -1 for locks\n");
19          scanf("%d", &locks);
20  }
            printf("\n total locks = %d\", tlocks);
21          printf("total stocks =%d\n", tstocks);
22
```

```
23          printf("total barrels =%d\n", tbarrels);

24          lsales = lprice*tlocks;
25          ssales = sprice*tstocks;
26          bsales = bprice*tbarrels;
27          sales = lsales + ssales + bsales;
28          printf("\n the total sales=%f\n", sales);
29          if(sales > 1800.0)
30          {
31                              comm=0.10*1000.0;
32                              comm=comm+0.15*800;
33                              comm=comm+0.20*(sales-1800.0);
          }
34          else if(sales > 1000)
35          {
36                              comm =0.10*1000;
37                              comm=comm+0.15*(sales-1000);
             }
38          else
39                    { comm=0.10*sales;
40                              }
41          printf" \n value of commission is\n");
42                     printf("the commission is=%f\n", comm);
43          return 0;  }
```

Define /Use nodes for variables in the commission problem

Variable name	Defined at node	Used at Node
lprice	7	24
sprice	8	25
bprice	9	26
tlocks	10,16	16, 21, 24
tstocks	11,17	17, 22, 25
tbarrels	12,18	18, 23, 26
locks	13,19	14,16
stocks	15	17
barrels	15	18
lsales	24	27
ssales	25	27
bsales	26	27
sales	27	28, 29, 33, 34, 37, 39
comm	31, 32, 33, 36, 37, 39	32, 33, 37, 42

RNSIT

Selected Define/Use Paths for Commission problem					
Test case id	**Description**	**Variables Path(Beginning, End nodes)**	**Du Paths**	**Definition clear?**	**Comments**
1	Check for lock price variable **DEF(lprice,7)** and **USE(lprice,24)**	(7 , 24)	<7-8-9-10-11-12-13-14-15-16-17-18-19-20-14-21-22-23-24>	Yes	
2	Check for Stock price variable **DEF(sprice,8)** and **USE(sprice,25)**	(8 , 25)	<8-9-10-11-12-13-14-15-16-17-18-19-20-14-21-22-23-24-25>	Yes	
3	Check for barrel price variable **DEF(bprice,9)** and **USE(bprice,26)**	(9 , 26)	<9-10-11-12-13-14-15-16-17-18-19-20-14-21-22-23-24-25-26>	Yes	
4	Check for total locks variable **DEF(tlocks,10) and DEF(tlocks,16)** and **3 usage nodes USE(tlocks,16), USE(tlocks,21), USE(tlocks,24)**	(10 , 16)	<10-11-12-13-14-15-16>	Yes	
		(10 , 21)	<10-11-12-13-14-15-16-17-18-19-20-14-21>	No	
		(10 , 24)	<10-11-12-13-14-15-16-17-18-19-20-14-21-22-23-24>	No	
		(16 , 16)	<16-16>	Yes	
		(16 , 21)	<16-17-18-19-14-21>	No	
		(16 , 24)	<16-17-18-19-20-14-21-22-23-24>	No	
5	Check for total stocks **variable DEF(tstocks,11) and DEF(tstocks,17)** and **3 usage nodes (USE(tstocks,17), USE(tstocks,22), USE(tstocks,25)**	(11 , 17)	<11-12-13-14-15-16-17>	Yes	
		(11 , 22)	<11-12-13-14-15-16-17-18-19-20-14-21-22>	No	
		(11, 25)	<11-12-13-14-15-16-17-18-19-20-14-21-22-23-24-25>	No	
		(17 , 17)	<17-17>	Yes	
		(17 , 22)	<17-18-19-20-14-21-22>	No	
		(17 , 25)	<17-18-19-20-14-21-22-23-24-25>	No	

6	check for locks variable **DEF(locks,13), DEF(locks,19) and USE(locks,14), USE(locks,16)**	(13 , 14)	<13-14>	Yes	Begin the loop
		(13 , 16)	<13-14-15-16>	Yes	
		(19 , 14)	<19-20-14>	Yes	
		(19 , 16)	<19-20-14-15-16>	Yes	Repeat the loop
7	Check for stocks variable **(DEF(stocks,15)** and **USE(stocks,17)**	(15 , 17)	<15-16-17>	Yes	
8	Check for sales variable **DEF(sales, 27)** and **USE(Sales, 28), USE(Sales , 29), USE(Sales,33) , USE(Sales , 34) , USE(Sales,37) , USE(Sales , 39)**	(27 ,28)	<27-28>	Yes	
		(27 , 29)	<27-28-29>	Yes	
		(27 , 33)	<27-28-29-30-31-32-33>	Yes	
		(27 , 34)	<27-28-29-34>	Yes	
		(27 , 37)	<27-28-29-34-35-36-37>	Yes	
		(27 , 39)	<27-28-29-34-38-39>	Yes	
9	Check for Commission variable **DEF(comm, 31,32,33) , DEF(comm,36,37) and DEF(comm,39)** and **USE(comm,42)**	((31,32,33),42)	<31-32-33-42>	Yes	
		((36 , 37) , 42)	<36-37-42>	Yes	
		(39 , 42)	<39 - 42>	Yes	

RNSIT

6. Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

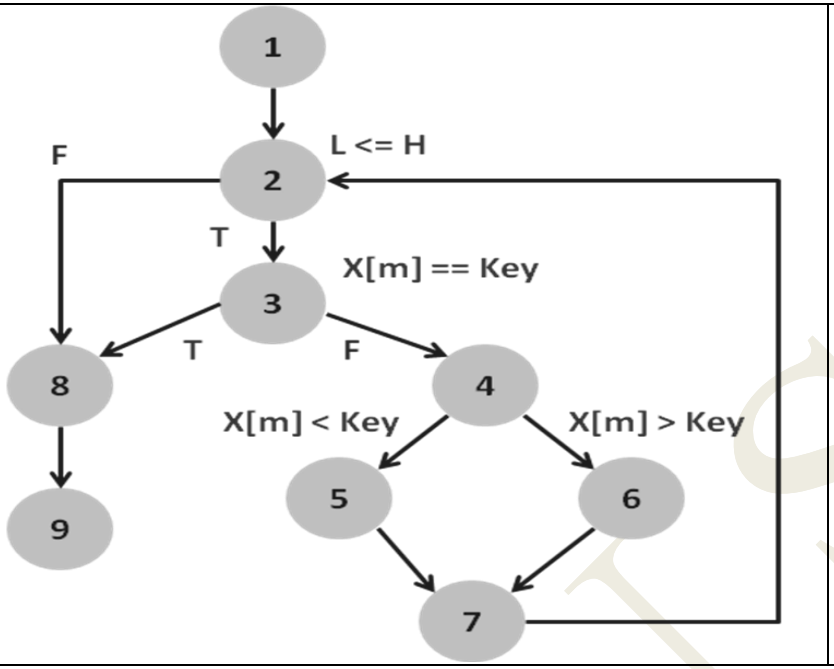
```
#include<stdio.h>
int binsrc(int x[],int low,int high,int key)
{
        int mid;
        while(low<=high)
        {
                mid=(low+high)/2;
                if(x[mid]==key)
                        return mid;
                if(x[mid]<key)
                        low=mid+1;
                else
                        high=mid-1;
        }
        return -1;
}

int main()
{
        int a[20],key,i,n,succ;
        printf("Enter the n value");
        scanf("%d", &n);
        if(n>0)
                {
        printf("enter the elements in ascending order\n");
                  for(i=0;i<n;i++)
                  scanf("%d", &a[i]);
```

RNSIT

```
        printf("enter the key element to be searched\n");
            scanf("%d",&key);

 succ=binsrc(a,0,n-1,key);
                if(succ>=0)
                        printf("Element found in position = %d\n", succ+1);
                else
                        printf("Element not found \n");
        }
        else
                printf("Number of element should be greater than zero\n");
        return 0;
}


int binsrc(int x[],int low, int high, int key)
{
        int mid;                                        1
        while(low<=high)                                2
                {
                mid=(low+high)/
                if(x[mid]==key)                         3
                        return mid;                     8
                if(x[mid]<key)                          4
                        low=mid+1;                      5
                        else
                        high=mid-1;                     6


        }                                               7
                return -1;                              8
}                                                       9
```

RNSIT

Program Graph – for Binary Search



Independent Paths:

#Edges=11, #Nodes=9, #P=1

V(G)= E-N+2P = 11-9+2 = **4**

P1: 1-**2**-3-8-9

P2: 1-**2**-3-4-5-7-**2**

P3: 1-**2**-3-4-6-7-**2**

P4: 1-**2**-8-9

RNSIT

Pre-Conditions/Issues:

Array has Elements in Ascending order	**T/F**
Key element is in the Array	**T/F**
Array has ODD number of Elements	**T/F**

Test Cases – Binary Search

Paths	Inputs		Expected Output	Remarks
	X[]	Key		
P1: 1-2-3-8-9	{10,20,30,40,50}	30	Success	Key ∈ X[] and Key==X[mid]
P2: 1-2-3-4-5-7-2	{10,20,30,40,50}	20	Repeat and Success	Key < X[mid] Search 1st Half
P3: 1-2-3-4-6-7-2	{10,20,30,40,50}	40	Repeat and Success	Key > X[mid] Search 2nd Half
P4: 1-2-8-9	{10,20,30,40,50}	60 OR 05	Repeat and Failure	Key ∉ X[]
P4: 1-2-8-9	Empty	Any Key	Failure	Empty List