

I N D E X

Particulars of the Experiments Performed

S.No.	Name of the EXPERIMENTS	Page	Date of Experiment	Date of Submission	Remarks
1	Boundary Value Analysis for triangle problem	1	7-11-22	5-12-22	(11) ✓
2	Boundary Value Analysis for Next Date problem	3	14-11-22	5-12-22	(10) ✓
3	Boundary Value Analysis for Commission problem	8	28-11-22	5-12-22	(10) ✓
4	Equivalence Class Partitioning for Next Date problem	12	5-12-22	2-1-23	(10) ✓
5	Equivalence Class Partitioning for triangle problem	17	2-1-23	9-1-23	(10) ✓
6	Binary Search algorithm using Path Testing	19	2-1-23	9-1-23	(10) ✓
7	Quick Sort algorithm using Path Testing	21	2-1-23	9-1-23	(10) ✓
8	Equivalence Class Partitioning for Commission Problem	23	2-1-23	9-1-23	(10) ✓
9	Absolute Letter Grading using Path Testing	27	9-1-23		

I N D E X

Particulars of the Experiments Performed

Test Case Name : Boundary Value Analysis for triangle problem
 Experiment Number : 1

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 - Boundary Value Test cases for input data

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comments
		a	b	c				
1	Enter the min value for a, b and c	1	1	1	Should display the message Equilateral Triangle.	Equilateral Triangle	PASS	$a = b = c$
2	Enter the min value for 2 items & min+1 for any one item.	1	1	2	Message should be displayed can't form a triangle.	Not a Triangle	PASS	$c > a + b$
3	Enter the min value for 2 items & min+1 for any one item	1	2	1	Message should be displayed can't form a triangle.	Not a Triangle	PASS	$a > b + c$
4	Enter the min value for 2 items & min+1 for any one item	2	1	1	Message should be displayed can't form a triangle.	Not a Triangle	PASS	$b > a + c$
5	Enter normal value for 2 items & 1 item is min value.	5	5	1	Should display the message Isosceles triangle.	Isosceles Triangle	PASS	$a = b$
6	Enter normal value for 2 items & 1 item is min value.	5	1	5	Should display the message Isosceles triangle.	Isosceles Triangle	PASS	$a = c$

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 triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Derive test cases for your program based on boundary value analysis, 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(" %i %i %i ", &a, &b, &c);
        printf("\n a=%i \t b=%i \t c=%i ", 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=%i is not the
range of permitted value!", a);
        if(!c2)
```

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comments
		a	b	c				
7	Enter normal value for 2 items & 1 item is min value for a, b and c	1	5	5	Should display the message Isosceles triangle.	Isosceles Triangle	PASS	$b=c$
8	Enter the normal value for 2 items & 1 item is max value for a, b and c	5	5	5	Should display message Equilateral triangle.	Equilateral Triangle	PASS	$a=b=c$
9	Enter normal value for 2 items & 1 item is max value for a, b and c	5	5	10	Should display message Not a triangle.	Not a Triangle	PASS	$c \neq a+b$
10	Enter normal value for 2 items & 1 item is max value for a, b and c	5	10	5	Should display message Not a triangle.	Not a Triangle	PASS	$b \neq a+c$
11	Enter normal value for 2 items & 1 item is max value for a, b and c	10	5	5	Should display message Not a triangle.	Not a Triangle	PASS	$a \neq b+c$
12	Enter the max value for 2 items & max-1 for any 1 item.	10	10	9	Should display the message Isosceles Triangle.	Isosceles Triangle	PASS	$a=b$
13	Enter the max value for 2 items & max-1 for any 1 item.	10	9	10	Should display the message Isosceles Triangle.	Isosceles Triangle	PASS	$a=c$
14	Enter the max value for 2 items & max-1 for any 1 item.	9	10	10	Should display the message Isosceles Triangle.	Isosceles Triangle	PASS	$b=c$
15	Enter the max value for a, b and c	10	10	10	Should display the message Equilateral Triangle.	Equilateral Triangle	PASS	$a=b=c$

Name of Experiment.....
Experiment No.....

Date.....
Experiment Result.....

Page No

2

printf("In The value of b+c is not the
range of permitted value!", b);

if (!c3)

printf("In The value of c+a is not the
range of permitted value!", c);

{ while (1 (c3 && c2 && c3));

if (a < b + c && b < a + c && c < a + b)

istriangle = 'y' ;

else

istriangle = 'n' ;

if (istriangle == 'sy')

if ((a == b) && (b == c))

printf(" Equilateral Triangle ");

else if ((a != b) && (a != c) && (b != c))

printf(" Scalene Triangle ");

else

printf(" Isosceles Triangle ");

return 0;

q

b. 1/2

q1

Test case name : Boundary Value Analysis test cases for Next date program
 Experiment Number : 2

Test data : Enter the three integer value

Pre-condition : Month 1 to 12 , Day 1 to 31 and year 1812 to 2013

Brief Description :

	Min	Min+1	Normal	Max-1	Max
Month	1	2	6	11	12
Day	1	2	15	30	31
Year	1812	1813	1912	2012	2013

Next date Output Boundary Value Analysis Cases

Case No	Description	Input Data			Expected Output			Actual Output			Status	Comments
		Month	Day	Year	Month	Day	Year	Month	Day	Year		
1	Enter the min value for month, day and year	1	1	1812	1	2	1812	1	2	1812	PASS	
2	Enter min+1 value for day, max value for year and min value for month.	1	2	2013	1	3	2013	1	3	2013	PASS	
3	Enter normal value for day, max-1 value for year and min value for month.	1	15	2012	1	16	2012	1	16	2012	PASS	

Name of Experiment (BVA) Next Date Problem Date 14 - 11 - 2022

Experiment No..... 2

Experiment Result.....

Page No.

3

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

```
#include <stdio.h>
int check (int day, int month)
{
    if ((month == 4 || month == 6 || month == 9 || month == 11)
        && day == 31)
        return 1;
    else
        return 0;
}
int isleap (int year)
{
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
        return 1;
    else
        return 0;
}
int main()
{
    int day, month, year, tomm-day, tomm-month, tomm-year,
        char flag;
```

Case No	Description	Input Data			Expected Output			Actual Output			Status Correctness	
		Month	Day	Year	Month	Day	Year	Month	Day	Year		
4	Enter normal value for day, max value for year, min for month	1	15	2013	1	16	2013	1	16	2013	PASS	
5	Enter max -1 value for day, normal for year, min for month	1	30	1912	1	31	1912	1	31	1912	PASS	
6	Enter max value for day, max for year, min for month	1	31	2013	2	1	2013	2	1	2013	MISS	
7	Enter max value for day, max for year and max for month	12	31	2013	1	1	2014	1	1	2014	PASS	
8	Enter max for day and min for year & month	1	31	1812	2	1	1812	2	1	2012	PASS	
9	Enter max -1 for day max -1 for year and max -1 for month	14	30	2012	12	1	2012	12	1	2012	PASS	
10	Enter min +1 for day and year and min for month	1	2	1813	1	3	1813	1	3	1813	PASS	

Name of Experiment _____ Date _____
Experiment No. 2 Page No. 6
Experiment Result _____

do

{

flag = 'y';

printf("Enter the today's date in the form of
dd mm yyyy\n");

scanf("%d-%d-%d", &day, &month, &year);

tomorrow-month = month;

tomorrow-year = year;

if (day < 1 || day > 31)

{

printf("value of day, not in range 1...31\n");

flag = 'n';

}

if (month < 1 || month > 12)

{

printf("value of month, not in range 1...12\n");

flag = 'n';

{

else if (check(day, month))

{

printf("value of day, not in range day <= 30\n");

flag = 'n';

{

if (year < 1812 || year > 2013)

{

printf("value of year, not in range 1812...2013\n");

flag = 'n';

{

Name of Experiment _____
Experiment No. 2

Date _____
Experiment Result _____

Page No. 5

```
if (month == 2)
{
    if (isLeap(year) && day > 29)
    {
        printf("invalid date input for leap year");
        flag = 'n';
    }
    else if (!isLeap(year)) && day > 28)
    {
        printf("invalid date input for not a leap year");
        flag = 'n';
    }
    while (flag == 'n');
    switch (month)
    {
        case 1 :
        case 3 :
        case 5 :
        case 7 :
        case 8 :
        case 10: if (day < 31)
                    tmm_day = day + 1;
                    else
                    {
                        tmm_day = 1;
                        tmm_month = month + 1;
                    }
                    break;
```

Name of Experiment _____
Experiment No. 2

Date _____
Experiment Result _____

Page No. 6

case 4 :

case 6 :

case 9 :

case 11 : if (day < 30)

 tomorrow-day = day + 1 ;

 else

 {

 tomorrow-day = 1 ;

 tomorrow-month = month + 1 ;

 }

 break ;

case 12 : if (day < 31)

 tomorrow-day = day + 1 ;

 else

 {

 tomorrow-day = 1 ;

 tomorrow-month = 1 ;

 if (year == 2013)

 {

 printf("the next day is out of boundary
 value of year \n");

 tomorrow-year = year + 1 ;

 }

 else

 tomorrow-year = year + 1 ;

 }

 break ;

Name of Experiment.....
Experiment No..... 2

Date.....
Experiment Result.....

Page No. 2

case 2 : if (day < 28)

tomm-day = day + 1 ;

else if (isLeap(year) && day == 28)

tomm-day = day - 1 ;

else if (day == 28 || day == 29)

{

tomm-day = 1 ;

tomm-month = 3 ;

}

break ;

9

printf("next day is : %d %d %d", tomm-day ,
tomm-month , tomm-year);

return 0 ;

3

✓

Test Case Name : Boundary Value for Commission Problem
 Experiment Number : 3

Test data : Price Rs for lock - 45.0, stock - 30.0 and barrel - 25.0

Sales = total lock * lock price + total stock * stock price + total barrel * barrel price
 Commission : 10% up to sales RS 1000, 15% off the next RS 800 and

20% on any sales in excess of 1800

Pre-condition : lock = -1 to exit and $1 \leq \text{lock} \leq 70$, $1 \leq \text{Stock} \leq 80$, $1 \leq \text{barrel} \leq 90$

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

Checking boundary value for locks, stocks and barrels and commission

Commission problem output boundary value analysis cases

Case Id	Description	Input Data			Expected Output		Actual Output		Status	Comments
		Total Locks	Total Stocks	Total Barrels	Sales	Commission	Sales	Commission		
1	Enter the min value for locks, stocks & barrels	1	1	1	100	10	100	10	Pass	Output minimum
2	Enter the min value from 2 items and min+1 for any one item	1	2	1	125	12.5	125	12.5	Pass	Output min+
3		1	2	1	130	13	130	13	Pass	Output min+
4		2	1	2	145	14.5	145	14.5	Pass	Output min +

Name of Experiment BVA for Commission Problem
 Experiment No. 3 Date 28-11-2022 Page No. 8

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

Assumption price for Lock = 45, stock = 30, barrels = 25 production limit could sell in a month 70 locks, 80 stocks, 90 barrels, commission on sales = 10% ≤ 1000 and 15% on 1000 to 1800 and 20% on above 1800.

```
#include < stdio.h >
int main()
{
    int locks, stocks, barrels, 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("Enter the number of locks and to exit\n"
           "the loop enter -1 for locks");
    scanf("%d", &locks);
    while (locks != -1)
    {
        if (locks > 70)
            locks = 70;
        if (stocks > 80)
            stocks = 80;
        if (barrels > 90)
            barrels = 90;
        sales = locks * lprice + stocks * sprice + barrels * bprice;
        if (sales <= 1000)
            comm = sales * 0.1;
        else if (sales > 1000 && sales <= 1800)
            comm = sales * 0.15;
        else
            comm = sales * 0.2;
        printf("Sales = %f, Comm = %f", sales, comm);
        tlocks += locks;
        tstocks += stocks;
        tbarrels += barrels;
        locks = 0;
        stocks = 0;
        barrels = 0;
    }
}
```

Case ID	Description	Input Data			Expected Output		Actual Output		Status	Comments
		Total Stocks	Total Barrels	Sales	Commission	Sales	Commission			
5.	Enter the value sales approx, mid value b/w 100 to 1000	5	5	5	500	50	500	50	Pass	mid-point
6.	Enter the values to calculate commission for sales nearly less than 1000	10	10	9	975	97.5	975	97.5	Pass	Border point
7.	Enter the value sales exactly equal to 1000	10	10	10	1000	100	1000	100	Pass	Border Point
8.	Enter max value for 2 items and max-1 for any one item.	70	80	89	7775	1415	7775	1415	Pass	Output max-
9.	Enter max value for stocks, stocks and barrels	70	80	90	7800	1420	7800	1420	Pass	Output max
10.	Enter random values such that to calculate commission for sales > than 1000	18	19	17	1805	321	1805	321	Pass	Random point

Name of Experiment.....
Experiment No..... 3

Date
Experiment Result.....

Page No

9

```
c1 = (locks <= 0 || locks > 70);
printf("\n Enter the number of stocks and barrels:");
scanf("%d %d", &locks, &barrels);
c2 = (stocks <= 0 || stocks > 80);
c3 = (barrels <= 0 || barrels > 90);
if (c1)
    printf("\n value of locks not in range 1....70");
else
{
    temp = locks + stocks;
    if (temp > 70)
        printf("\n new total locks = %d not in the
range 1....70 so old ", temp);
    else
        locks = temp;
}
printf("\n total locks = %d ", locks);
if (c2)
    printf("\n value of stocks not in the range 1...80");
else
{
    temp = stocks + temp;
    if (temp > 80)
        printf("\n new total stocks = %d not in the
range 1....80 so old ", temp);
    else
        stocks = temp;
}
```

Name of Experiment _____
Experiment No. 3

Date _____
Experiment Result _____

Page No. 10

```
printf("In total stocks = .1.d ", tstocks);
if (c3)
    printf("In value of barrels not in the range 1...90");
else
{
    temp = tbarrels + barrels;
    if (temp > 90)
        printf("In new total barrels = .1.d not in the
range 1...90 so old ", temp);
    else
        tbarrels = temp;
}
printf("In total barrels = .1.d ", tbarrels);
printf(" In Enter the number of locks and to exit the
loop enter -1 for locks : ");
scanf(".1.d", &locks);
if (locks > 0 && tstocks > 0 && tbarrels > 0)
{
    sales = lprice * tlocks + sprice * tstocks + bprice * tbarrels;
    printf("In the total sales = .1.f ", sales);
    if (sales > 1800.0)
    {
        comm = 0.10 * 1000.0;
        comm = comm + 0.15 * 800;
    }
}
```

Name of Experiment.....

Date

Page No.

11

Experiment No.....

Experiment Result.....

comm = comm + 0.20 * (sales - 1800.0);
}

else if (sales > 1000)

{

comm = 0.10 * 1000;

comm = comm + 0.15 * (sales - 1000);

}

else

comm = 0.10 * sales;

printf("The commission of = %.f ", comm);

}

else

printf("Commission cannot be calculated ");

return 0;

}

By
S. K. Jaiswal

Test case Name : Equivalence class test cases for Next Date
Experiment Number : 4

Test Data : Enter the three integer value

Pre-condition : Month 1 to 12 , Day 1 to 31 , year 1812 to 2013

Valid Cases

$$M1 = \{ \text{month} : 1 \leq \text{month} \leq 12 \}$$

$$D1 = \{ \text{day} : 1 \leq \text{day} \leq 31 \}$$

$$Y1 = \{ \text{year} : 1812 \leq \text{year} \leq 2013 \}$$

Invalid cases

$$M2 = \{ \text{month} : \text{month} < 1 \}$$

$$M3 = \{ \text{month} : \text{month} > 12 \}$$

$$D2 = \{ \text{day} : \text{day} < 1 \}$$

$$D3 = \{ \text{day} : \text{day} > 31 \}$$

$$Y2 = \{ \text{year} : \text{year} < 1812 \}$$

$$Y3 = \{ \text{year} : \text{year} > 2013 \}$$

Design and develop a program in a language of your choice to solve the Next Date Function. Analyze it from the perspective of equivalence class value testing, derive different test cases, execute these test cases and discuss the test results.

```
#include <stdio.h>

int check (int day, int month)
{
    if ((month == 4 || month == 6 || month == 9 || month == 11) &&
        day == 31)
        return 1;
    else
        return 0;
}

int isleap (int year)
{
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
        return 1;
    else
        return 0;
}

int main()
{
    int day, month, year, tomm-day, tomm-month, tomm-year;
    char flag;
    do
    {

```

Next Date Output Equivalence Class Testing

C 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		
WNI, SNI	Enter the M1, D1 and Y1 valid cases	6	15	1912	6	16	1912	6	16	1912	PASS	valid case

C weak Robustness Equivalence Class)

Case Id	Description	Input Data			Expected Data			Actual Output			Status	Comment
		month	day	year	month	day	year	month	day	year		
WR1	Enter M1, D1 and Y1 cases	6	15	1912	6	16	1912	6	16	1912	PASS	
WR2	Enter M2, D1 and Y1 cases	-1	15	1912	Should display month not in range 1...12							Value of month not in range 1..12
WR3	Enter M1, D3 and Y1 cases	6	32	1912	Should display day not in range 1..31							Value of day not in range 1...31
WR4	Enter the M1, D1 and Y2 cases	6	15	1811	Should display year not in range 1812...2013							Value of year not in range 1812...2013
WR5	Enter M1, D1 and Y3 cases	6	15	2014	Should display year not in range 1812...2013							Value of year not in range 1812...2013

```
flag = 'y';
printf("Enter the today's date in the form of
dd mm yyyy: ");
scanf("%d %d %d", &day, &month, &year);
tempmonth = month;
tempyear = year;
if (day < 1 || day > 31)
{
    printf("value of day, not in range 1...31");
    flag = 'n';
}
if (month < 1 || month > 12)
{
    printf("value of month, not in range 1...12");
    flag = 'n';
}
else if (check(day, month))
{
    printf("value of day, not in range day <= 30");
    flag = 'n';
}
if (year <= 1812 || year > 2013)
{
    printf("value of year, not in range 1812...2013");
    flag = 'n';
}
if (month == 2)
```

C Strong Robustness Equivalence Class

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comment
		month	day	year				
SR1	Enter M2, D1 and Y1 cases	-1	15	1912	Should display value of month not in range 1..12	Value of month not in range 1...12	PASS	
SR2	Enter M1, D2 and Y1 cases	6	-1	1912	Should display value of day not in range 1...31	Value of day not in range 1...31	PASS	
SR3	Enter M2, D2 and Y1 cases	-1	-1	1912	i) Should display value of month not in range 1...12 ii) Should display value of day not in range 1...31	Value of month not in range 1...12 Value of day not in range 1...31	PASS PASS	
SR4	Enter M2, D1 and Y2 cases	-1	15	1811	i) Should display month not in range 1...12 ii) Should display year not in range 1812...2013	Value of month not in range 1...12 Value of year not in range 1812...2013	PASS PASS	

Name of Experiment.....
Experiment No..... 4

Date.....
Experiment Result.....

Page No.

14

```
if ( isleap (year) && day > 29 )
{
    printf (" invalid date input for leap year " );
    flag = 'n' ;
}

else if ( ! ( isleap (year) ) && day > 28 )
{
    printf (" invalid date input for not a leap year " );
    flag = 'n' ;
}

while ( flag == 'n' );
switch (month)
{
    case 1 :
    case 3 :
    case 5 :
    case 7 :
    case 8 :
    case 10 : if ( day < 31 )
                tomm-day = day + 1 ;
                else
                {
                    tomm-day = 1 ;
                    tomm-month = month + 1 ;
                }
                break ;
}
```

(Some addition equivalence boundary checking)

Case Id	Description	Input Data			Expected Output			Actual Output			Status	Comment
		day	month	year	day	month	year	day	month	year		
1	Enter D1, M1 and Y1 cases	31	12	1811	Should display year not in range 1812...2013			Value of year not in range 1812...2013			PASS	
2	Enter the D1, M1 and Y2 cases	31	12	2012	1	1	2013	1	1	2013	PASS	
3	Enter D1, M1 and Y3 valid cases	31	12	2013	Should display next date is out of boundary 2013			Next day is out of boundary, value for next day is 1 1 2014			PASS	
4	Enter D1, M1 and Y1 cases	2	29	2011	Should display invalid input			Invalid date input for not a leap year			PASS	
5	Enter D2, M2 and Y1 cases	-2	-7	1912	i) Should display month not in range 1...12 ii) Should display day not in range 1...31			Value of month not in range 1...12			PASS	
6	Enter D1, M1, Y1 cases	28	28	2004	29	2	2004	29	2	2004	PASS	

Name of Experiment.....

Date.....

Page No

15

Experiment No.....

Experiment Result.....

case 4 :

case 6 :

case 9 :

case 11 : if (day < 30)

tomm-day = day + 1 ;

else

{

tomm-day = 1 ;

tomm-month = month + 1 ;

{

break ;

case 12 : if (day < 31)

tomm-day = day + 1 ;

else

{

tomm-day = 1 ;

tomm-month = 1 ;

if (year == 2013)

{

printf(" the next day is out
of boundary value of year ");

tomm-year = year + 1 ;

{

else

tomm-year = year + 1 ;

{

break ;

Name of Experiment _____
Experiment No. 4

Date _____
Experiment Result _____

Page No. 16

CASE 2 : if (day < 23)

tomm-day = day + 1 ;

else if (isLeap(year) == day == 25)

tomm-day = day + 1 ;

else if (day == 23 || day == 29)

{

tomm-day = 1 ;

tomm-month = 3 ;

}

break ;

}

printf(" next day is : %d-%d-%d ", tomm-day ,
tomm-month , tomm-year) ;

return 0 ;

}

Test Case Name : Equivalence class analysis for triangle problem

Experiment Number : 5

Test Data : Enter the 3 integers value (a, b & c)

Classes : { (1,3), (3,5), (5,7), (7,10) }

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

Brief-condition description : Check whether given value for a triangle or can't form a triangle.

Triangle Problem - Equivalence Class Test Cases for I/P data

(Weak Equivalence Class Testing)

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comment
		a	b	c				
1	From class(1,3)	1	1	1	Should display Equilateral Triangle	Equilateral Triangle	PASS	$a = b = c$
2	From class(3,5)	3	5	3	Should display Isosceles Triangle	Isosceles Triangle	PASS	$a = c$
3	From class(3,5)	3	4	5	Should display Scalene Triangle	Scalene Triangle	PASS	$a \neq b \neq c$
4	From class(3,5)	4	3	8	Should display Not a triangle	Not a triangle	PASS	$c \neq a + b$

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 triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle or they do not form a triangle at all. Derive test cases for your program based on Equivalence Class Partitioning and discuss the results.

```
#include <stdio.h>
int main()
{
    int a,b,c1,c2,c3;
    char istriangle;
    do
    {
        printf("\n Enter 3 integers which are sides of triangle : ");
        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);
```

C Weak Robust Equivalence Class Testing)

Case ID	Description	Input Data			Expected Output	Actual Output	Status	Comments
		a	b	c				
5	Enter one invalid ifp and 2 valid for a,b,c	-1	5	5	Should display value of a not in permitted value range	Value of a = -1 not in range	PASS	a = -1
6	Enter one invalid ifp & 2 valid for a,b,c	5	-1	5	Should display value of b not in permitted value range	Value of b = -1 not in range	PASS	b = -1
7	Enter one invalid input and 2 valid for a,b,c	5	5	11	Should display value of c not in range or permitted value	Value of C = 11 not in range	PASS	c = 11 c > 10

C Strong Robust Equivalence Class Testing)

8	Enter two invalid input and one valid for a,b,c	-1	-1	5	i) Should display value of a is not in range ii) Should display value of b is not in range	Value of a = -1 not in range Value of b = -1 not in range	PASS	a = -1 b = -1
9	Enter two invalid input and one valid for a,b,c	5	-1	-1	i) Should display value of b is not in range ii) Should display value of c is not in range	Value of b = -1 not in range Value of c = -1 not in range	PASS	b = -1 c = -1
10	Enter two invalid input and one valid for a,b,c	-1	5	-1	i) Should display value of a is not in range ii) Should display value of c is not in range	Value of a = -1 not in range Value of c = -1 not in range	PASS	a = -1 c = -1
11	Enter all invalid inputs	-1	-1	-1	i) a is not in range ii) Should display value of a = -1 is iii) Should display value of b = -1 is iv) Should display value of c = -1 is	Value of a = -1 not in range Value of b = -1 not in range Value of c = -1 not in range	PASS	a = -1 b = -1 c = -1

Name of Experiment.....
Experiment No..... 5

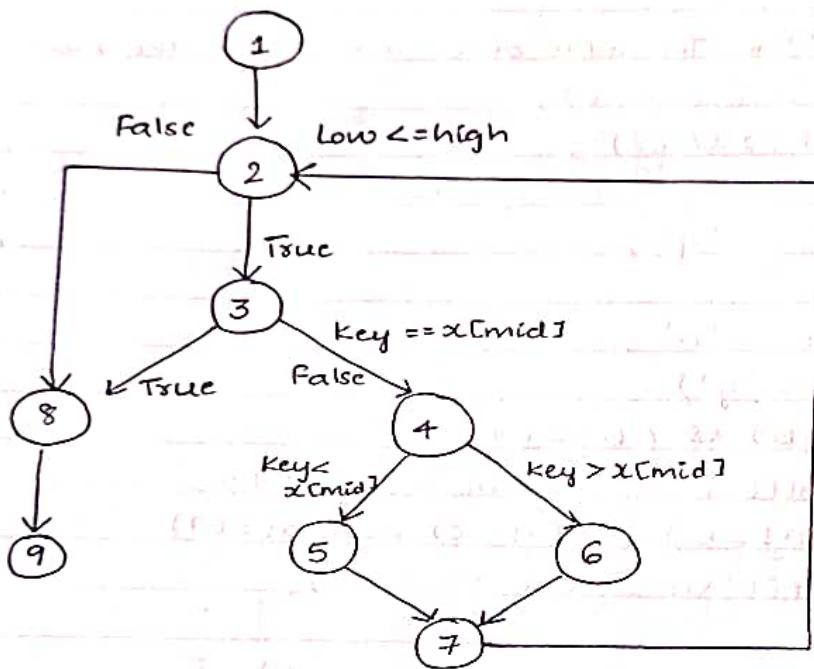
Date
Experiment Result.....

Page No 12

```
if (!c3)
    printf("In The value of c = -1. d is not the range of
permitted value ", c);
g while (!(c1 && c2 && c3));
if (a < b + c && b < a + c && c < a + b)
    istriangle = 'y';
else
    istriangle = 'n';
if (istriangle == 'y')
    if ((a == b) && (b == c))
        printf("In Equilateral Triangle ");
    else if ((a != b) && (a != c) && (b != c))
        printf("In Scalene Triangle ");
    else
        printf("In Isosceles Triangle ");
else
    printf("In Not a triangle ");
return 0;
```

3

Program Graph for Binary Search



Independent Paths :-

$$\# \text{Edges} = 11, \# \text{Nodes} = 9, \# P = 1$$

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

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

Design, develop a 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;
```

1

```
    while (low <= high)
```

2

{

```
        mid = (low + high) / 2;
```

3

```
        if (key == x[mid])
```

4

```
            return mid;
```

5

```
        if (key < x[mid])
```

6

```
            high = mid - 1;
```

```
        else
```

7

```
            low = mid + 1;
```

8

{

```
    return -1;
```

9

{

```
int main()
```

{

```
    int x[20], key, i, n, succ;
```

```
    printf("In Enter the n value : ");
```

```
    scanf("%d", &n);
```

```
    if (n > 0)
```

{

Pre - Conditions / Issues :

Array has elements in ascending order T/F

Key element is in the Array T/F

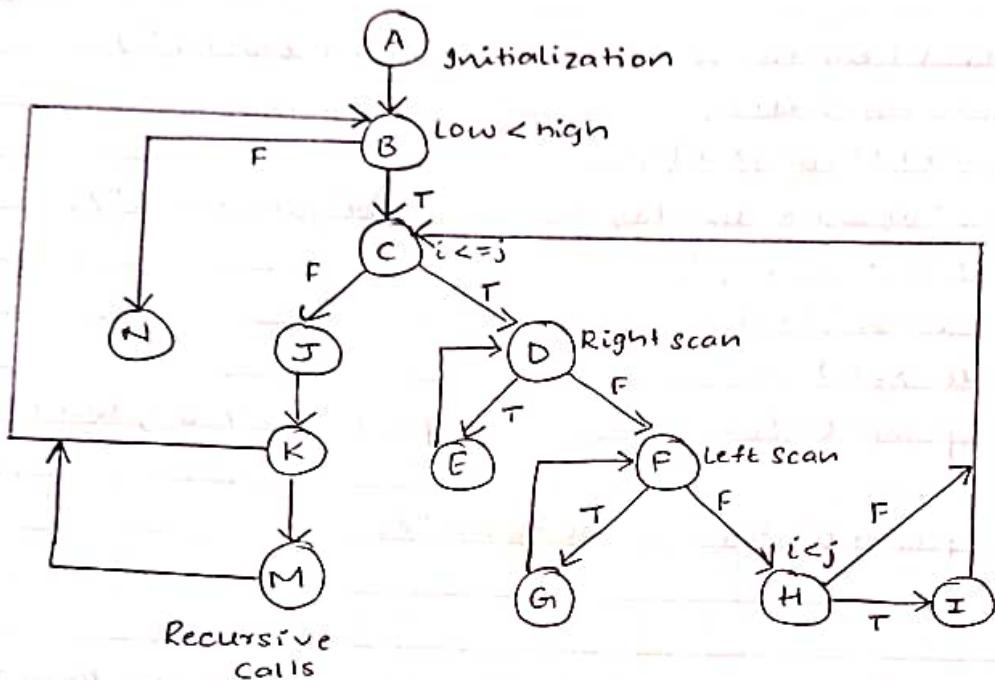
Array has odd number of elements T/F

Test Cases - Binary Search

Paths	Inputs		Expected Output	Actual Output	Status	Remarks
	X[]	Key				
P1 : 1-2-3-8-9	{1,2,3,4,5}	3	Success	Element found in position 3	PASS	key == x[mid]
P2 : 1-2-3-4-5-7-2	{1,2,3,4,5}	2	Repeat & Success	Element found in position 2	PASS	key < x[mid] Search 1st half
P3 : 1-2-3-4-6-7-2	{1,2,3,4,5}	4	Repeat & Success	Element found in position 4	PASS	key > x[mid] Search 2nd half
P4 : 1-2-8-9	{1,2,3,4,5}	6	Repeat & Failure	Element not found	PASS	key != x[]
P4 : 1-2-8-9	Empty	Any Key	Failure	Failure	PASS	Empty List

```
printf("In Enter the elements in ascending order : ");
for(i=0; i<n; i++)
    scanf("%f", &x[i]);
printf("In Enter the key element to be searched : ");
scanf("%f", &key);
succ = binSrch(x, 0, n-1, key);
if (succ >= 0)
    printf(" Element found in position = %d ", succ+1);
else
    printf(" Element not found ");
}
else
    printf(" In Number of element should be greater than zero");
return 0;
}
```

Program Graph - Quick Sort



Independent Paths - Quick Sort

P1: A - B - N

P2: A - B - C - J - N - B

P3: A - B - C - J - K - M - B

P4: A - B - C - D - F - H - C

P5: A - B - C - D - F - H - I - C

P6: A - B - C - D - E - D - F - H

P7: A - B - C - D - F - G - F - H

Independent Paths:

$$\# \text{Edges} = 18, \# \text{Nodes} = 13, \# P = 1$$

$$V(G) = E - N + 2P = 18 - 13 + 2 = 7$$

Name of Experiment..... Quick Sort.....
Experiment No..... 7.....

Date 02 - 01 - 2023
Experiment Result.....

Page No. 21

Design, develop, code and run the program in any suitable language to implement the Quick-Sort Algorithm. Determine the basic paths and using them derive different test cases, execute the test cases and discuss the test results.

```
#include <stdio.h>
void quick-sort (int a[100], int low, int high)
{
    int i, j, temp, key ;
    if (low < high)
    {
        key = a[low] ;
        i = low ;
        j = high ;
        while (i <= j)
        {
            while (key >= a[i])
                i++ ;
            while (key < a[j])
                j-- ;
            if (i < j)
            {
                temp = a[i] ;
                a[i] = a[j] ;
                a[j] = temp ;
            }
        }
    }
}
```

1 A
2 B
3
4
5
6 C
7 D
8 E
9 F
10 G
11 H
12 I
13
14

Pre - Conditions / Issues :

Array has only one Element , Two Elements , Three Elements (6 Possibilities)
Array has elements in ASC / DSC / Arbitrary

Ex : 3 elements : 123 , 132 , 213 , 312 , 321 , 221 , 111 , 333

Test Cases - Quick Sort

Paths	Inputs $X[]$	First Last	Expected Output	Actual Output	Status	Remarks
P1 : A - B - N	5	1,1	Sorted	5	PASS	Only one element
P2 : A - B - C - J - K - B	5,4	1,2	Repeat & Sorted	4,5	PASS	Two elements
P3 : A - B - C - J - K - M - B	1,2,3 OR 3,1,2	1,3	Repeating sorted	1,2,3 err	PASS	Three elements
P4 : A - B - C - D - F - H - C	1,2,3,4,5	1,5	Repeat & Sorted	1,2,3,4,5	PASS	ASC Seq
P5 : A - B - C - D - F - H - I - C	5,4,3,2,1	1,5	Repeat & Sorted	1,2,3,4,5	PASS	DSC Seq
P6 : A - B - C - D - E - D - F - H	1,4,3,2,5 OR 2,2,2,2,2	1,5	Repeat & Sorted	1,2,3,4,5	PASS	First 5 min
P7 : A - B - C - D - F - G - F - H	5,2,3,1,4	1,5	Repeat & Sorted	1,2,3,4,5	PASS	First 5 max
P8 : A - B - N	Empty	Any key	Failure	Failure	Pass	NC is empty

Name of Experiment.....
Experiment No..... 7

Date
Experiment Result.....

Page No. 22

```
temp = a[low];  
a[low] = a[j];  
a[j] = temp;  
quickSort(a, low, j-1);  
quickSort(a, j+1, high);
```

15 J

16 K

17 M

18

19

3

20 N

```
void main()
```

```
{
```

```
int i, n, a[100];
```

```
printf("\n Enter the value for n : ");
```

```
scanf(" %d ", &n);
```

```
for (i=0; i<n; i++)
```

```
scanf(" %d ", &a[i]);
```

```
quickSort(a, 0, n-1);
```

```
printf("\n The sorted array is : ");
```

```
for (i=0; i<n; i++)
```

```
printf(" %d ", a[i]);
```

```
}
```

Test Case Name : Equivalence Class for Commission Problem
 Experiment Number : 8

Test data : price Rs for locks - 45.0 , stock - 30.0 and barrel - 25.0
 $\text{Sales} = \text{total locks} * \text{lock price} + \text{total stock} * \text{stock price} + \text{total barrel} * \text{barrel price}$
 commission = 10% . up to sales Rs 1000 , 15% of the next Rs 800 and 20% on any sales in excess of 1800.

Pre-condition : locks = -1 to exit and $1 \leq \text{locks} \leq 70$, $1 \leq \text{stock} \leq 80$ & $1 \leq \text{barrel} \leq 90$

Brief- Description : The salesperson had to sell at least one complete rifle per month.
 Checking boundary value for locks , stocks , barrels and commission .

Valid cases

L1 = { locks : $1 \leq \text{locks} \leq 70 }$
 L2 = { locks = -1 }

Invalid cases

L3 = { locks : locks = 0 or locks $< -1 }$
 L4 = { locks : locks $> 70 }$
 S2 = { stocks : stocks $< 1 }$

S3 = { stocks : stocks $> 80 }$
 B2 = { barrels : barrels $< 13 }$
 B3 = { barrels : barrels $> 90 }$

Commission Problem Output equivalence class Testing

C weak & Strong Normal Equivalence Class)

Case Id	Description	Input Data			Expected Output		Actual Output		Status
		Total Locks	Total Stocks	Total Barrels	Sales	Commission	Sales	Commission	
1	Enter value within range for locks,stocks and barrels	35	40	45	3900	640	3900	640	Pass

Name of Experiment E.C.T for Commission
Experiment No. 8

Problem

Date 02 - 01 - 2023

Experiment Result

Page No. 23

Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of equivalence class, derive 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 could sell in a month 70 locks, 80 stocks and 90 barrels commission on sales = 10%. <= 1000 and 15% on 1000 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("Enter the number of locks and to exit the loop
enter -1 for locks ");
    scanf("-1.%d", &locks);
    while (locks != -1)
    {
```

Weak Robustness Equivalence Class

Case Id	Description	Input Data			Expected Output	Actual Output	Status
		Locks	Stocks	Barrels			
WR1	Enter the value of locks = -1	-1	40	45	Terminates i/p drop and proceeds to calculate stated conn		PASS
WR2	Enter value less than -1 / equal 0 Locks and other valid inputs	0	40	45	Value of locks not in range 1...70	Value of locks not in range 1...70	PASS
WR3	Enter value of stocks greater than 50 and others valid inputs	35	81	45	Value of stocks not in range 1...80	Value of stocks not in range 1...80	PASS
WR4	Enter value less than 1 equal 0 barrels and other valid i/p	35	40	0	Value of barrels not in range 1...90	Value of barrels not in range 1...90	PASS

Strong Robustness Equivalence class)

Case Id	Description	Input Data			Expected Output	Actual Output	Status
		Locks	Stocks	Barrels			
SRI	Enter the value less than -1 for locks & others valid inputs	-2	40	45	Value of locks not in range 1...70	Value of locks not in range 1...70	PASS
SR2	Enter locks & stocks less than equal to 0 and others valid i/p	-2	-1	45	Value of locks not in range Value of stocks not in range	Value of locks not in range 1...70 Value of stocks not in range 1...80	PASS
SR3	Enter stocks & barrels less than equal to 0 and others valid i/p	35	-1	-1	Value of stocks not in range Value of barrels not in range	Value of stocks not in range 1...80 Value of barrels not in range 1...90	PASS
SR4	invalid inputs	-2	-2	-2	Value of locks, stocks and barrels not in range	Locks not in range stocks not in range barrels not in range	PASS

```
c1 = (locks <= 0 || locks > 70);
printf("\n Enter the number of stocks and barrels.");
c2 = (stocks <= 0 || stocks > 80);
c3 = (barrels <= 0 || barrels > 90);
if (c1)
    printf ("\n Value of locks not in range 1...70");
else
    {
        temp = tlocks + locks;
        if (temp > 70)
            printf ("\n New total locks = %d not in range
1...70 so old ", temp);
        else
            tlocks = temp;
    }
printf ("\n Total locks = %d\n", tlocks);
if (c2)
    printf ("\n Value of stocks not in range 1...80");
else
    {
        temp = fstocks + stocks;
        if (temp > 80)
            printf ("\n New total stocks = %d not in range
1...80 so old ", temp);
        else
            fstocks = temp;
    }
printf ("\n Total Stocks = %d", fstocks);
```

C Some addition equivalence boundary checking)

Case Id	Description	Input Data			Expect Output		Actual Output		Status
		Locks	Stocks	Barrels	Sales	Commission	Sales	Commission	
OR1	Enter the value for locks; stocks, barrels where 0<sales <1000	5	5	5	500	50	500	50	PASS
OR2	Enter the value for locks, stocks, barrels where 1000 < sales <1800	15	15	15	1500	175	1500	175	PASS
OR3	Enter the value for locks, stocks, barrels where sales >1800	25	25	25	2500	360	2500	360	PASS

Name of Experiment.....
Experiment No..... 8

Date
Experiment Result.....

Page No 25

```
if(c3)
    printf("\n Value of barrels not in the range 1...90");
else
{
    temp = tbarrels + barrels;
    if(temp > 90)
        printf("\n New total barrels = %d not in the
range 1...90 so old ", temp);
    else
        tbarrels = temp;
}
printf("\n Total barrel = %d", tbarrels);
printf("\n Enter the number of locks and to exit the
loop enter -1 for locks :");
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 + hprice * tbarrels;
printf ("\n The total sales = %f ", sales);
if(tlocks > 0 && tstocks > 0 && tbarrels > 0)
{
    if(sales > 1800)
    {
        comm = 0.10 * 1000.0;
        comm = comm + 0.15 * 800;
        comm = comm + 0.20 * (sales - 1800);
    }
}
```

Name of Experiment.....
Experiment No.....

Date.....
Experiment Result.....

Page No. 26

else if (sales > 1000)

{

comm = 0.10 * 1000 ;

comm = comm + 0.15 * (sales - 1000) ;

}

else

comm = 0.10 * sales ;

printf ("The commission is = %.f ", comm) ;

}

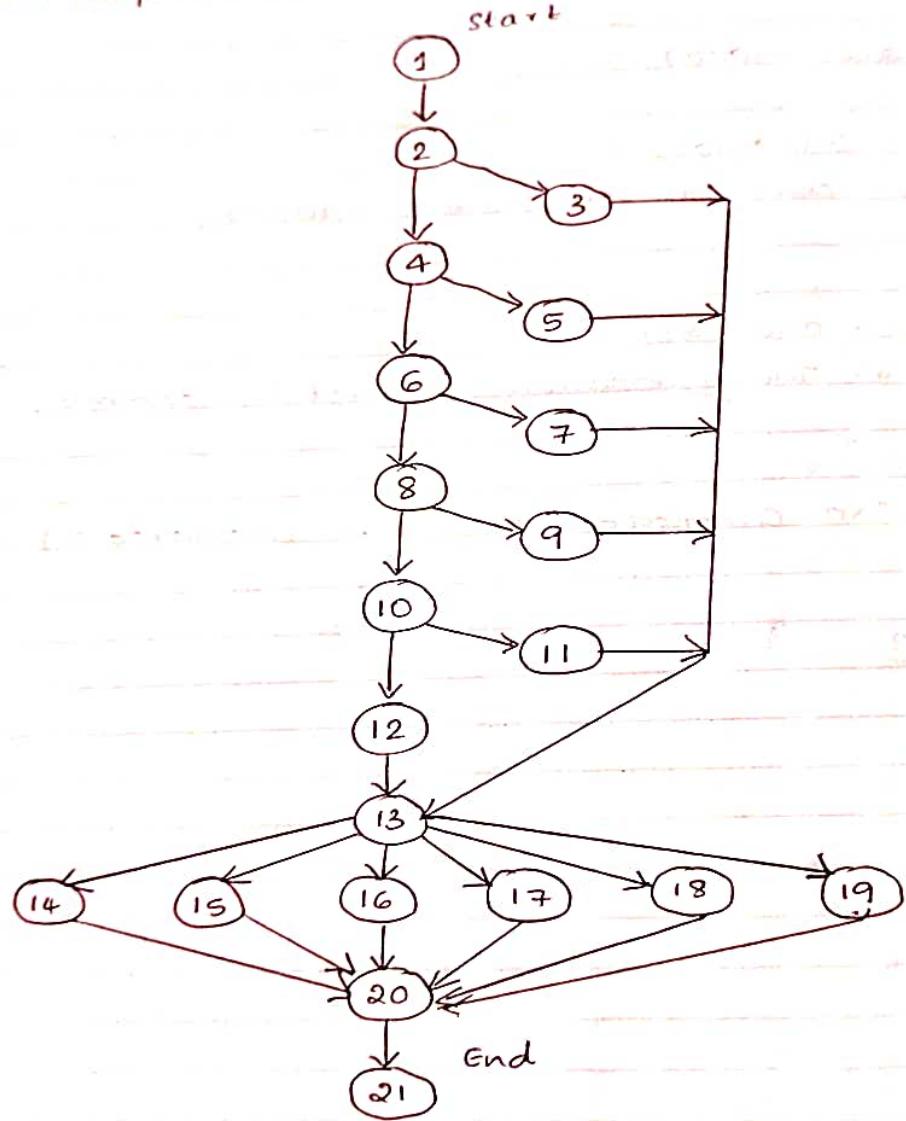
else

printf ("Commission cannot be calculated ") ;

return 0 ;

}

Program Graph



Edges : 26 # Nodes = 189 # P = 1

$$V(G) = E - N + 2P = 26 - 19 + 2 = 09$$

Design, develop and code, run the program in any suitable language to implement an absolute letter grading procedure, making suitable assumptions. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

```
#include < stdio.h >
int main()
{
    float per;
    char grade;
    scanf ("%.f", &per);
    if (per >= 90)
        grade = 'A';
    else if (per >= 80 && per < 90)
        grade = 'B';
    else if (per >= 70 && per < 80)
        grade = 'C';
    else if (per >= 60 && per < 70)
        grade = 'D';
    else if (per >= 35 && per < 60)
        grade = 'E';
    else
        grade = 'F';
    switch (grade)
    {
        case 'A' : printf ("In Excellent");
                    break;
```

Pre-conditions / Issues :- Percentage 'Per' is a positive float number

Test Cases - Absolute Grading

Percentage 'Per' is a positive float number

Paths	Input Per	Expected Output	Actual Output	Remarks
P1 : 1-2-4-6-8-10-11-17 -19-20	<60	22.5	F Grade, Fail	Percentage is 22.500000 Grade is F PASS
P2 : 1-2-4-6-8-9-11-16 -19-20	60-69	64	D grade, Above average	Above Average Percentage is 64.000000 Grade is D Good
P3 : 1-2-4-6-7-11-15-19 -20	70-79	78	C grade, Good	Percentage is 78.000000 Grade is C Very Good
P4 : 1-2-4-5-11-15-20-21	80-89	82.678	B grade, Very Good	Percentage is 82.678000 Grade is B Excellent
P5 : 1-2-3-13-14-20-21	>=90	98.5	A grade, Excellent	Percentage is 98.500000 Grade is A Fail
P6 : 1-2-4-6-8-10-11-13 -14-20-21	<60	12.89	Excellent	Percentage is 12.890000 Grade is F FAIL
P7 : 1-2-4-6-8-10-12 -13-16-20-21	<60	24.02	Good	Percentage is 24.200000 Grade is F FAIL
P8 : 1-2-4-6-8-10-12 -13-17-20-21	<60	46.6	Above average	Satisfactory Percentage is 46.600000 Grade is E FAIL
P9 : 1-2-4-6-8-10-12 -13-19-20-21	<60	59.9	Fail	Satisfactory Percentage is 59.900000 Grade is E FAIL

Name of Experiment.....
Experiment No..... 9

Date
Experiment Result.....

Page No. 28

```
case 'B' : printf ("\n Very Good"); 15
    break;
case 'C' : printf ("\n Good"); 16
    break;
case 'D' : printf ("\n Above average"); 17
    break;
case 'E' : printf ("\n Satisfactory"); 18
    break;
default : printf ("\n Fail"); 19
    break;
```

g
printf ("\n The percentage = .1.f and grade is .1-c ",
per, grade); 20
return 0; 21

}

Test Case Name : Decision Table for triangle problem

Experiment Number : 10

Test Data : Enter the 3 integer Values (a, b, c)

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

Brief-Description: Check whether given value for equilateral , isosceles and scalene triangle or cannot 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
	C2 : $b < a + c$	-	F	T	T	T	T	T	T	T	T
	C3 : $c < a + b$	-	-	F	T	T	T	T	T	T	T
	C4 : $a = b$	-	-	-	T	T	T	F	F	F	F
	C5 : $a = c$	-	-	-	T	T	F	T	T	F	F
	C6 : $b = c$	-	-	-	T	F	T	F	T	F	F
Actions	a1 : Not a Triangle	X	X	X							
	a2 : Scalene Triangle									X	
	a3 : Isosceles Triangle								X	X	
	a4 : Equilateral Triangle						X				
	a5 : Impossible						X	X	X		

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 triangle and determine if the three values represent an equilateral triangle, isosceles triangle or they do not form a triangle at all. Derive test cases for your program based on decision-table approach. Execute test cases and discuss the results.

```
#include <stdio.h>
int main()
{
    int a,b,c;
    char istriangle;
    printf("In Enter 3 integers which are sides of a
triangle : ");
    scanf("%d %d %d", &a, &b, &c);
    printf("In a = %d , b = %d & c = %d ", a, b, c);
    if (a < b + c && b < a + c && c < a + b)
        istriangle = 'y';
    else
        istriangle = 'n';
    if (istriangle == 'y'):
        if (a == b) && (b == c))
            printf("In Equilateral triangle");
        else if ((a != b) && (a != c) && (b != c))
            printf("In Scalene triangle");
        else
            printf("In Isosceles triangle");
}
```

Case Id	Description	Input Data			Expected Output	Actual Output	Status	Comments
		a	b	c				
1	Enter the values of a,b,c such that 'a' is not less than other 2 sides.	8	5	2	Message should be displayed can't form a triangle.	Not a Triangle	PASS	$a \neq b+c$
2	Enter the values of a,b,c such that 'b' is not less than other 2 sides.	4	12	3	Message should be displayed can't form a triangle.	Not a Triangle	PASS	$b \neq a+c$
3	Enter the value of a,b,c such that 'c' is not less than a and b	2	4	22	Message should be displayed can't form a triangle.	Not a Triangle	PASS	$c \neq a+b$
4	Enter the value a,b,c satisfying precondition $a=b$, $b=c$, $c=a$	12	12	12	Should display the message as Equilateral Triangle	Equilateral Triangle	PASS	$a=b=c$
5	Enter the value of a,b,c satisfying pre-condition and $a=b$, $b \neq c$	5	5	9	Should display the message as Isosceles Triangle	Isosceles Triangle	PASS	$a=b$ and $b \neq c$
6	Enter the values of a,b,c - satisfying precondition & $a \neq b \neq c$	2	4	5	Should display the message as Scalene Triangle	Scalene Triangle	PASS	$a \neq b \neq c$
7	Other Inputs	-1	2	4	Should display Not a Triangle	Not a Triangle	PASS	$a < -1$

Name of Experiment.....
Experiment No..... 10

Date.....
Experiment Result.....

Page No. 20

printf ("in Isosceles Triangle");
else

printf ("in Not a Triangle");
return 0;

3

Name of Experiment DT for Commission
Experiment No. 11

Problem Date 14-01-2023

Experiment Result.....

Page No. 31

Design, Develop, Code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of decision table, derive test cases, execute these test cases and discuss the test results.

Assumption price for locks = 45.0, stock = 30.0 and barrels = 25.0 production limit could sell in a month 70 locks, 80 stocks and 90 barrels commission on sales = 10% <= 1000 and 15% on 1000 to 1800 and 20% on above 1800.

```
#include < stdio.h >
int main()
{
    int locks, stocks, barrels, tlocks, tstocks, tbarrels;
    float lprice, sprice, bprice, sales and comm;
    int c1, c2, c3, temp;
    lprice = 45.0;
    sprice = 30.0;
    bprice = 25.0;
    tlocks = 0;
    tstocks = 0;
    tbarrels = 0;
    printf("Enter the no. of locks and to exit the loop
enter -1 for locks : ");
    scanf(" %d ", &locks);
    while (locks != -1)
```

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

Rules		R1	R2	R3	R4
Condition	C1: stocks > 0 & & $commissions \geq 0$	T	T	T	F
	C2: sales > 0 and sales ≤ 1000	T	F	F	
	C3: sales > 1001 and sales ≤ 1800		T	F	
	C4: sales ≥ 1801			T	
Actions	A1: Cannot calculate commission			X	
	A2: comm = 10% * sales	X			
	A3: comm = 10% * 1000 + (sales - 1000) * 15%		X		
	A4: comm = 10% * 1000 + 15% * 800 + (sales - 1800) * 20%				X

Name of Experiment.....
Experiment No..... 1.1

Date
Experiment Result.....

Page No. 32

```
c1 = (locks <= 0 || locks > 70);  
printf("In Enter the number of stocks and barrels : ");  
scanf(" %d %d ", &stocks, &barrels);  
c2 = (stocks <= 0 || stocks > 80);  
c3 = (barrels <= 0 || barrels > 90);  
if (c1)  
    printf("In Value of locks not in the range 1...70");  
else  
{  
    temp = tlocks + locks;  
    if (temp > 70)  
        printf("In new total locks = %d not in the  
        range 1...70 so old ", temp);  
    else  
        tlocks = temp;  
}  
printf("In total locks = %d ", tlocks);  
if (c2)  
    printf("In Value of stocks not in the range 1...80");  
else  
{  
    temp = tstocks + stocks;  
    if (temp > 80)  
        printf("In new total stocks = %d not in the  
        range 1...80 so old ", temp);  
    else  
        tstocks = temp;  
}
```

Pre-condition : Initial value total locks = 0 , total stocks : 0 and Total Barrels = 0

Pre-condition Limit : Total locks , stocks and barrels should not exceed limit 10, 80, 90

NIR - Not in Range

SAE0 - Same as Expected Output

Commission Problem - Decision Table Test cases for Input Data

Case Id	Description	Input Data			Output			Actual Output	Total Stocks	Total Barrels	Status
		Locks	Stocks	Barrels	Terminate i/p loop	Calculate Commission					
1	Enter the value locks :-1				Yes	0		SAE0			PASS
2	Enter valid input for lock and stock and invalid for barrels.	20	30	-5	No				20	30	NIR
3	Enter valid input for locks and barrels and invalid for stocks	15	-2	45	No				15	45	SAE0
4	Enter valid input for stocks and barrels and invalid for locks	-4	15	16	No				15	16	SAE0
5	Enter valid input for stocks and invalid for barrels	15	89	100	No				15	NIR	PASS
6	Enter valid input for stocks and invalid for locks and barrels	88	20	125	No				20	NIR	SAE0
7	Enter valid input for barrels and invalid for locks and stocks	100	200	25	No				25	SAE0	PASS

Name of Experiment.....
Experiment No.....

Date

Experiment Result.....

Page No. 33

```
printf("In Total stocks = %.d ", tstocks);
if (c3)
    printf("in Value of barrels not in range 1...90 ");
else
{
    temp = tbarrels + barrels;
    if (temp > 90)
        printf("In new total barrels = %.d not in the
range 1...90 so old ", temp);
    else
        tbarrels = temp;
}
printf("In Total barrel = %.d ", tbarrels);
printf("In Enter the no. of locks and to exit the loop
enter -1 for locks : ");
scanf("%d", &locks);
printf("In Total locks = %.d In total stocks = %.d In total
barrels = %.d ", tlocks, tstocks, tbarrels);
sales = lprice * tlocks + sprice * tstocks + bprice * tbarrels;
printf("In The total sales = %.f ", sales);
if (tlocks > 0 && tstocks > 0 && tbarrels > 0)
{
    if (sales > 1800.0)
        comm = 0.10 * 1000.0;
    comm = comm + 0.15 * 800;
```

Commission Problem - Decision Table Test cases for commission calculation

L = Locks

S = Stocks

B = Barrels

Pre-condition: Locks = -1

Case Id	Description	Input Data			Expected Output			Actual Output	Status
		L	S	B	Expected Sales	Actual Sales	Expected Commission Commissi.on		
1	Check the value of Sales = 0	0	0	0	0	0	0	SAEO	PASS
2	If sales value within these range (sales > 0 and sales ≤ 1000)	10	9	10	970	970	97	97	SAEO PASS
3	If sales value within these range (sales > 1000 and sales ≤ 1800)	15	20	13	1600	1600	160	160	PASS SAEO
4	If sales value within these range (sales > 1800)	30	20	55	3325	3325	525	525	SAEO PASS

Name of Experiment.....
Experiment No..... 11

Date
Experiment Result.....

Page No. 34

comm = comm + 0.20 * (sales - 1800.0);

{

else if (sales > 1000)

{

comm = 0.10 * 1000 ;

comm = comm + 0.15 * (sales - 1000) ;

}

else

comm = 0.10 * Sales ;

printf("The commission is = %.f ", comm);

{

else

printf(" Commission cannot be calculated : ");

return 0;

}

Test Case Name : Data flow Testing for Commission Problem
 Experiment No : 12

Pre condition : Enter -1 for locks to exit input loop

Brief Description : Enter the locks , stocks and barrels > 0

Define / Use nodes for variables in the commission problem

Variable Name	Defined at Node	Used at Node
lprice	8	54
sprice	9	57
bprice	10	57
tlocks	11 , 29	25, 31 , 56, 57
tstocks	12 , 40	36, 42 , 56, 57
tbarrels	13 , 51	47 , 53 , 56, 57
locks	14,	15, 17 , 25 , 54
stocks	18	19 , 36
barrels	18	20 , 47
sales	57	58 , 61 , 65 , 67 , 70 , 73
comm	63 , 64 , 65 , 69 , 70 , 73	64 , 65 , 70 , 74
temp	25 , 36 , 47	26 , 27 , 29 , 37 , 38 , 40 , 48 , 49 , 51
c1	17	21
c2	19	32
c3	20	43

Name of Experiment DGT for Commission Problem Date 14 - 01 - 2023
Experiment No. 18 Experiment Result

Page No. 35

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 test results.

1 Program 12

```
2 #include <stdio.h>
3 int main()
4 {
5     int locks, stocks, barrels, tlocks, tstocks, tbarrels;
6     float lprice, sprice, bprice, sales, comm;
7     int c1, c2, c3, temp;
8     lprice = 45.0;
9     sprice = 30.0;
10    bprice = 25.0;
11    tlocks = 0;
12    tstocks = 0;
13    tbarrels = 0;
14    printf("In Enter the number of locks and to exit the
15        loop enter -1 for locks :");
16    scanf("%-1d", &locks);
17    while (locks != -1)
18    {
19        c1 = (locks <= 0 || locks > 70);
20        printf("In Enter the number of stocks and barrels :");
21        scanf("%-1d %-1d", &stocks, &barrels);
22        c2 = (stocks <= 0 || stocks > 80);
23        c3 = (barrels <= 0 || barrels > 90);
```

Selected Define / Use Paths for Commission Problem

Test Case Id	Description	Variables Path (Beginning, End) Nodes	DU Paths	Definitions Create ?	Common -erors
1	Check for lock price variable DEF(lprice, 8) and USE(lprice, 57)	(8, 57)	< 8 - 9 - 10 - 11 - ... - 55 - 56 - 57 >	Yes	No other defining node in path
2	Check for stock price variable DEF(sprice, 9) and USE(sprice, 57)	(9, 57)	< 9 - 10 - 11 - 12 - ... - 55 - 56 - 57 >	Yes	No other defining node in path
3	Check for barrel price variable DEF(bprice, 10) and USE(bprice, 57)	(10, 57)	< 10 - 11 - 12 - ... - 55 - 56 - 57 >	Yes	No other defining node in path
4	Check for total locks variable DEF(tlocks, 11) and DEF(tlocks, 29) and # usage nodes USE(tlocks, 25), USE(tlocks, 31), USE(tlocks, 56), USE(tlocks, 57)	(11, 25) (11, 31) (11, 56) (11, 57) (29, 25) (29, 31)	< 11 - 12 - 13 - 14 - ... - 21 - 22 - 23 - 24 - 25 > < 11 - 12 - 13 - 14 - ... - 30 - 31 > < 11 - 12 - 13 - 14 - ... - 29 - 30 - 31 > < 11 - 12 - 13 - 14 - ... - 29 - 55 - 56 > < 11 - 12 - 13 - 14 - ... - 29 - 55 - 56 > < 29 - 30 - 31 >	Yes No No No No Yes	" Node 29 is a defining node No other defining node in path

Name of Experiment.....
Experiment No.....

Date.....
Experiment Result.....

Page No. 36

```
21 if (c1)
22     printf (" Value of locks not in the range 1...70 ");
23 else
24 {
25     temp = tlocks + locks ;
26     if (temp > 70)
27         printf ("\n New total locks = -1-d not in the
range 1...70 so old ", temp );
28     else
29         tlocks = temp;
30 }
31 printf (" total locks = -1-d \n ", tlocks);
32 if (c2)
33     printf (" Value of stocks not in the range 1...80 ");
34 else
35 {
36     temp = tstocks + stocks ;
37     if (temp > 80)
38         printf (" new total stocks = -1-d not in the range
1...80 so old ", temp );
39     else
40         tstocks = temp;
41 }
42 printf (" total stocks = -1-d \n ", tstocks);
43 if (c3)
44     printf (" Value of barrels not in the range 1...90 ");
45 else
```

Test Case Id	Description	Variables Path (Beginning, End) Node	Do Paths	Definition clean?	Comm -ents
5	Check for checks variable DEF(stocks, 18) and USE(stocks, 19), USE(stocks,36)	(18, 19) (18 , 36)	<18 - 19 7 <18 - 19 - 20 - 21 - ... - ... - - 36 >	Yes Yes	No other defining node No other defining node
6	Check for Sales DEF(57) & USE(sales, 58), USE(sales, 61), USE(sales,65), USE(sales, 67), USE(sales,70), USE(sales, 73)	(57, 58) (57, 61) (57, 65) (57, 67) (57, 70) (57, 73)	<57 , 58> <57 - 58 - 59 - 60 - 61> <57 - 58 - 59 - 60 - 64-65> <57 - 58 - 66 - 67> <57 - 58 - 69 - 70> <57 - 58 - - - - 72-73>	Yes Yes Yes Yes Yes Yes	No - other defining node for sales
7	Check for commission DEF(comm, 63,64,65), DEF(comm,69,70), DEF(comm , 73) and USE(comm,740)	((63,64,65), 74, ((69,70), 740)) (73 , 74)	<63 - 64 - 65 - 74 > < 69 - 70 - 74 > < 73 - 74 >	Yes Yes Yes	11

Name of Experiment.....
Experiment No..... 12

Date
Experiment Result.....

Page No. 37

```
46    {
47    temp = tbarrels + barrels;
48    if (temp > 90)
49        printf ("new total barrels = %.d not in the range
50           1...90 so old", temp);
51    else
52        tbarrels = temp;
53    }
54    printf ("total barrel = %.d", tbarrels);
55    printf ("In Enter the number of locks and to exit the
56    loop enter -1 for locks");
57    scanf ("%d", &locks);
58    }
59    printf ("In total locks = %d In total stocks = %d In total
60    barrels = %.d", locks, tstocks, tbarrels);
61    sales = (lprice * locks + sprice * tstocks + bprice * tbarrels);
62    printf ("In The total sales = %.f In ", sales);
63    if (locks > 0 && tstocks > 0 && tbarrels > 0)
64    {
65        comm = 0.10 * 1000.0;
66        comm = comm + 0.15 * 800;
67        comm = comm + 0.15 * (sales - 1800.0);
68    }
```

Test case Id	Description	Variable Path (Beginning, End) Nodes	DU Paths	Definition Clear?	Comm -ent
8	Check for temp variable DEF (temp , 25) , DEF (temp, 26) , and DEF (temp, 47) and use (temp, 26) , use (temp, 27) and so on.	(25, 26) (36, 48) (47, 51) ⋮	< 25 - 26 > < 36 - 37 - 38 - 39 ... - 47 - 48 > < 47 - 48 - 49 - 50 - 51 >	Yes No Yes	No other defining node in path 47 is a defining node of temp No other defining node in path
9	Check for c1 variable DEF (c1, 17) and use (c1, 21)	(17, 21)	< 17 + 18 + 19 + 20 - 21 >	Yes	No other defining node for c1 in path
10	Check for c2 variable DEF (c2, 19) and use (c2, 32)		< 19 - 20 - 21 - 22 ... - 31 - 32 >	Yes	No other defining node for c2 in path

Name of Experiment.....
Experiment No.....

Date

Page No. 38

```
69 comm = 0.10 * 1000;  
70 comm = comm + 0.15 * (sales - 1000);  
71 }  
72 else  
73     comm = 0.10 * sales;  
74     printf ("The commission is = %.f ", comm);  
75 }  
76 else  
77     printf ("Commission cannot be calculated ");  
78 return 0;  
79 }
```