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.

/* 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);

                                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;
}
```

Test Case Name :Decision Table for Commission Problem

Experiment Number : 8

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	R5	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

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			