

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 the D1, M1 and Y1 valid cases	31	12	1811	Should display the message value of the year in range 1812..2013							
2	Enter the D1, M1 and Y2 valid cases	31	12	2012	1	1	2013					
3	Enter the D1, M1 and Y3 valid cases	31	12	2013	Should display the message Next is out of boundary 2013							

Program 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.

Program:

/* 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*/

Code:

```
#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("\nenter 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 so old ",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 so old ",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 so old ",temp);
else
tbarrels=temp;
}
printf("total barrel=%d",tbarrels);
printf("\nenter the number of locks and to exit the loop enter -1 for locks\n");
scanf("%d",&locks);
}
printf("\ntotal locks = %d\ntotal stocks =%d\ntotal barrels =%d\n",tlocks,tstocks,tbarrels); sales
= lprice*tlocks+sprice*tstocks+bprice*tbarrels;
printf("\nthetotal 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);
}
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 : 7

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 % of the next Rs 800 and 20 % on any sales in excess of 1800

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

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	R10
Conditions	C1: Locks = -1	T	F	F	F	F	F	F	F	F
	C2 : $1 \leq \text{Locks} \leq 70$	-	T	T	F	T	F	F	F	T
	C3 : $1 \leq \text{Stocks} \leq 80$	-	T	F	T	F	T	F	F	T
	C4 : $1 \leq \text{Barrels} \leq 90$	-	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
	a5 : Calculate Sales	X								
	a6: proceed to commission decision table	X								

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

RULES		R1	R2	R3	R4
Condition	C1 : Sales = 0	T	F	F	F
	C1 : Sales > 0 AND Sales ≤ 1000		T	F	F
	C2 : Sales > 1001 AND sales ≤ 1800			T	F
	C3 : sales ≥ 1801				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 lock and stack and invalid for barrels	20	30	-5	Total of locks, stocks is updated if it is with in a precondition limit and Should display value of barrels is not in the range 1..90			
3	Enter the valid input for lock and barrels and invalid for stocks	15	-2	45	Total of locks, barrels is updated if it is with in a precondition limit and Should display value of barrels is not in the range 1..80			
4	Enter the valid input for lock and barrels and invalid for stocks	-4	15	16	Total of stocks , barrels is updated if it is with in a precondition limit and Should display value of barrels is not in the range 1..70			
5	Enter the valid input for lock and invalid value for stocks and barrels	15	80	100	Total of locks is updated if it is with in 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 with in 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 with in 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 with in 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 with in these range(Sales > 0 AND Sales ≤ 1000)	900	Then commission = 0.10*sales = 90	900			
3	if sales value with in these range(Sales> 1000 AND Sales≤ 1800)	1400	Then commission = 0.10*1000 + 0.15*(sales - 1000)	1600			
4	if sales value with in these range(Sales > 1800	2500	Then commission = 0.10*1000 + 0.15*800 + 0.20 *(sales - 1800)	3400			

Program 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.

Code for Decision-Table Approach:

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