10	Enter one invalid input and two valid value for a , b and c	5	5	11	Should display value of a is not in the range of permitted values			
Strong Robust Equivalence class Testing								
11	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			
12	Enter one invalid input and two valid value for a , b and c	5	-1	5	Should display value of a is not in the range of permitted values			
13	Enter one invalid input and two valid value for a , b and c	5	5	-1	Should display value of a is not in the range of permitted values			
14	Enter two invalid input and two valid value for a , b and c	-1	-1	5	Should display value of a is not in the range of permitted values			
					Should display value of b is not in the range of permitted values			
14	Enter two invalid input and two valid value for a , b and c	5	-1	-1	Should display value of b is not in the range of permitted values			
					Should display value of c is not in the range of permitted values			
14	Enter two invalid input and two valid value for a , b and c	-1	5	-1	Should display value of a is not in the range of permitted values			
					Should display value of c is not in the range of permitted values			
15	Enter all invalid inputs	-1	-1	-1	Should display value of a is not in the range of permitted values			
					Should display value of b is not in the range of permitted values			
					Should display value of c is not in the range of permitted values			

Program 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.

```
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;
13 printf("\nenter the number of locks and to exit the loop enter -1 for locks\n");
scanf("%d", &locks);
14 while(locks!=-1) {
15 printf("enter the number of stocks and barrels\n");
scanf("%d%d",&stocks,&barrels);
16 tlocks=tlocks+locks;
17 tstocks=tstocks+stocks;
18 tbarrels=btarrels+barrels;
19 printf("\nenter the number of locks and to exit the loop enter -1 for  locks\n");
scanf("%d",&locks);
20 }
21 printf("\ntotal locks = %d\",tlocks);
22 printf("total stocks =%d\n",tstocks);
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("\nthe 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 printf("the commission is=%f\n",comm);
41 return 0;
42 }
```

Test Case Name : Data Flow Testing for Commission Program

Experiment No : 4

Precondition : Enter -1 for locks to exit from 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	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

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)	<7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24>	Yes	
2	Check for Stock price variable DEF(sprice,8) and USE(sprice,25)	(8 , 25)	<8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25>	Yes	
3	Check for barrel price variable DEF(bprice,9) and USE(bprice,26)	(9 , 26)	<9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26>	Yes	
4	Check for total locks variable DEF((tlocks,10) and DEF(tlocks,16)) and 3 usage node(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 node(USE(tstocks,17),USE(tstocks,22),USE(tstocks,25)	(11 , 17)	<11-12-13-14-15-16-17>	Yes	
		(11 , 22)	<11-12-13-14-15-16-17-18-19-20-21-14-21>	No	
		(11, 25)	<11-12-13-14-15-16-17-18-19-20-21-14-21-23-24-25>	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 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,34,35) and DEF(comm,39) and USE(comm,42)	((31,32,33),42)	<31-32-33-42>	Yes	
		((34 , 35) , 42)	<34-35-42>	Yes	
		((39 , 42)	<39 - 42>	Yes	