## SOFTWARE TESTING LABORATORY

Subject Code: 21ISL66 I.A. Marks: 50

Hours/Week: 03 Exam Hours: 03

Total Hours: 4 Exam Marks: 50

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.
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.
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.
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.
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.
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

**Program 1**

**(Boundary, Equivalence and Decision Test Case for Commission Problem)**

**/\* 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 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("\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)

else

{

printf("value of locks not in the range 1..70 ");

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)

else

{

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

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)

else

{

}

printf("value of barrels not in the range 1..90 ");

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("\nthe total sales=%f\n",sales);

if(tlocks>0&&tstocks>0&&tbarrels>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)

{

}

else

comm =0.10\*1000; comm=comm+0.15\*(sales-1000);

comm=0.10\*sales;

}

else

printf("the commission is=%f\n",comm);

printf(" Commission cannot be calculated \n");

return 0;

}

## Test Case Name : Boundary Value for Commission Problem Experiment Number : 1

**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< =lock < = 70 , 1<=stock <=80 and 1<=barrel<=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** |  |
| **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 | Enter the values sales exactly equal to 1000 | 10 | 10 | 10 | 1000 | 100 |  |  |  | Border point |
| 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 values normal value for lock, stock and  barrel | 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** | **Commissi on** | **Sales** | **Commi ssion** | **Status** | **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 + |

## PROGRAM 3

## Test Case Name :Decision Table 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 % 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** | **R 3** | **R4** | **R5** | **R 6** | **R7** | **R8** | **R10** |
| **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 |
| 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 : tlocks>0 && tstocks>0 && tbarrels>0 | T | T | T | F |
| C1 : Sales > 0 AND Sales ≤ 1000 | T | F | F |  |
| C2 : Sales > 1001 AND sales ≤ 1800 |  | T | F |  |
| C3 : sales ≥1801 |  |  | T |  |
| **Actions** | A1 : Cannot calculate the 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 |  |

**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 NIR-Not in Range**

**SAEO-Same as Expected Output**

**Commission Problem -Decision Table Test cases for input data**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cas e Id** | **Description** | **Input Data** | | | **Output** | | | | | **Actual Output** | **Status** | **Comments** |
| **Locks** | **Stocks** | **Barrels** | **Terminate i/p loop** | **Calculate commission** | **Total locks** | **Total stocks** | **Total Barrels** |
| 1 | Enter the value of Locks= -1 | -1 |  |  | yes | 0 |  |  |  | SAEO | PASS |  |
| 2 | Enter the valid input for lock and stack and invalid for barrels | 20 | 30 | -5 |  | no | 20 | 30 | NIR | SAEO | PASS |  |
| 3 | Enter the valid input for lock and barrels and invalid for stocks | 15 | -2 | 45 |  |  |  |  |  |  |  |  |
| 4 | Enter the valid input for lock and barrels and invalid for  stocks | -4 | 15 | 16 |  |  |  |  |  |  |  |  |
| 5 | Enter the valid input for lock and invalid value for stocks and barrels | 15 | 80 | 100 |  |  |  |  |  |  |  |  |
| 6 | Enter the valid input for stocks and invalid value for locks and barrels | 88 | 20 | 99 |  |  |  |  |  |  |  |  |
| 7 | Enter the valid input for  barrels and invalid value for locks and stocks | 100 | 200 | 25 |  |  |  |  |  |  |  |  |

**Commission Problem -Decision Table Test cases for commission calculation**

**L-locks S-Stocks**

**B-Barrels**

**Precondition : Locks = -1**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Case Id** | **Description** | **Input Data** | | | **Expected Output** | | | | **Actual Output** | **Status** | **Comments** |
| **L** | **S** | **B** | **Expecte Sales** | **Actual Sales** | **Expected Commission** | **Actual Commission** |
| 1 | Check the value of  sales=0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SAEO | PASS |  |
| 2 | if sales value with in  these range( Sales > 0 AND Sales ≤ 1000 ) | 10 | 9 | 10 |  |  |  |  |  |  |  |
| 3 | if sales value with in  these range( Sales > 1000 AND Sales ≤ 1800 ) | 15 | 15 | 15 |  |  |  |  |  |  |  |
| 4 | if sales value with in these range( Sales > 1800 | 20 | 30 | 40 |  |  |  |  |  |  |  |

**Program 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 analysis. Derive different test cases, execute these test cases and discuss the test results. \*/**

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

#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

{

flag='y';

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

scanf("%d%d%d",&day,&month,&year); tomm\_month=month;

tomm\_year= year; if(day<1 || day>31)

{

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

}

if(month<1 || month>12)

{

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

flag='n';

}

else if(check(day,month))

{

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

}

if(year<=1812 || year>2013)

{

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

flag='n';

}

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)

tomm\_day=day+1;

case 4:

case 6:

case 9:

else

{

}

break;

tomm\_day=1; tomm\_month=month+1;

case 11: if(day<30)

tomm\_day=day+1;

else

{

}

break;

tomm\_day=1; tomm\_month=month+1;

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\n"); tomm\_year=year+1;

case 2:

}

break;

}

else

tomm\_year=year+1;

if(day<28)

tomm\_day=day+1;

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

tomm\_day=day+1; else if(day==28 || day==29)

{

break;

}

tomm\_day=1; tomm\_month=3;

}

printf("next day is : %d %d %d",tomm\_day,tomm\_month,tomm\_year); return 0;}

## Test Case Name : Equivalence class test cases for Next date 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

**Valid Cases**

M1 = { month ; 1 ≤ month ≤ 12 }

D1 = { day : 1 ≤ day ≤ 31 }

Y1 = { year : 1812 ≤ year ≤ 2013 }

**Invalid cases** M2 = {month : month < 1} M3 = {month : month > 12} D2 = {day : day < 1}

D3 = {day : day > 31}

Y2 = {year : year < 1812} Y3 = {year : year > 2013}

**Next date Output 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 M1, D1 and Y1 valid cases | 6 | 15 | 1912 | 6 | 16 | 1912 |  |  |  |  |  |

**( Weak Robustness Equivalence Class )**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Case Id | **Description** | **Input Data** | | | **Expected Output** | | | **Actual output** | | | **Status** | **Comme**  **nt** |
| **month** | **day** | **year** | **month** | **day** | **year** | **mon**  **th** | **day** | **year** |  |  |
| WR1 | Enter the M1, D1 and Y1 cases | 6 | 15 | 1912 | 6 | 16 | 1912 |  |  |  |  |  |
| WR2 | Enter the M2 , D1 and Y1 cases | -1 | 15 | 1912 | Should display the message value  of the month not in the range 1..12 | | |  | | |  |  |
| WR3 | Enter the M3 ,D1 and Y1 cases | 13 | 15 | 1912 | Should display the message value of the month not in the range  1..12 | | |  | | |  |  |
| WR4 | Enter the M1, D2 and Y1 cases | 6 | -1 | 1912 | Should display the message value of the day not in the range 1..31 | | |  | | |  |  |
| WR5 | Enter the M1, D3 and Y1 cases | 6 | 32 | 1912 | Should display the message value of the day not in the range 1..31 | | |  | | |  |  |
| WR6 | Enter the M1, D1 and Y2 cases | 6 | 15 | 1811 | Should display the message value of the year not in the range  1812..2013 | | |  | | |  |  |
| WR7 | Enter the M1, D1 and Y3 cases | 6 | 15 | 2014 | Should display the message value of the year not in the range  1812..2013 | | |  | | |  |  |

## Program 4: Decision Table Approach for Solving Triangle Problem

**/\* 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 decision-table approach, execute the test cases and discuss the results \*/**

#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",a,b,c);

**// to check is it a triangle or not**

if( a<b+c && b<a+c && c<a+b ) istriangle='y';

else

istriangle ='n';

**// to check which type of triangle**

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

else

printf("isosceles triangle\n");

printf("Not a triangle\n"); return 0;

}

## 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 | Cant form a triangle | PASS | R1 (a < b+c condition fails) |
| 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 |  |  | R2 |
| 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 |  |  | R3 |
| 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 |  |  | R4(a=b=c) |
| 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 |  |  | R7(a=b) |
| 6 | Enter the value a, b and c satisfying precondition and  a ≠b , b ≠ c and c ≠ a | 5 | 6 | 7 | Should display the message Scalene triangle |  |  | R11(a!=b!=c) |

## Program 4 (Boundary Value and Equivalence Class Analysis Program)

**/\* 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("\nenter 3 integers which are sides of triangle\n"); scanf("%d%d%d",&a,&b,&c); printf("\na=%d\tb=%d\tc=%d",a,b,c);

c1 = a>=1 && a<=10; c2= b>=1 && b<=10; c3= c>=1 && c<=10; if (!c1)

printf("\nthe value of a=%d is not the range of permitted value",a);

if (!c2)

if (!c3)

printf("\nthe value of b=%d is not the range of permitted value",b);

printf("\nthe 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';

**// To check which type of triangle**

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

else

printf("isosceles triangle\n");

printf("Not a triangle\n"); return 0;

}

## Test Case Name :Boundary Value Analysis for triangle problem Experiment Number : 2

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

**Pre-condition : 1 ≤ a ≤ 10 , 1 ≤ b ≤ 10 and 1 ≤ c ≤ 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 | PASS | a=b=c |
| 2 | Enter the min value for 2 items and  min +1 for any one item1 | 1 | 1 | 2 | Message should be displayed can't form a  triangle |  |  |  |
| 3 | Enter the min value for 2 items and min +1 for any one item1 | 1 | 2 | 1 | Message should be displayed can't form a triangle |
| 4 | Enter the min value for 2 items and min +1 for any one item1 | 2 | 1 | 1 | Message should be displayed can't form a triangle |
| 5 | Enter the normal value for 2 items and 1 item is min value | 5 | 5 | 1 | Should display the message Isosceles triangle |  |  |  |
| 6 | Enter the normal value for 2 items and 1 item is min value | 5 | 1 | 5 | Should display the message Isosceles triangle |  |  |  |
| 7 | Enter the normal value for 2 items and 1 item is min value | 1 | 5 | 5 | Should display the message Isosceles triangle |  |  |  |
| 8 | Enter the normal Value for a, b and c | 5 | 5 | 5 | Should display the message Equilateral triangle |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | Enter the normal value for 2 items and 1 item is max value | 5 | 5 | 10 | Should display the message Not a triangle |  |  |  |
| 10 | Enter the normal value for 2 items and 1 item is max value | 5 | 10 | 5 | Should display the message Not a triangle |  |  |  |
| 11 | Enter the normal value for 2 items and 1 item is max value | 10 | 5 | 5 | Should display the message Not a triangle |  |  |  |
| 12 | Enter the max value for 2 items and max - 1 for any one item | 10 | 10 | 9 | Should display the message Isosceles triangle |  |  |  |
| 13 | Enter the max value for 2 items and max - 1 for any one item | 10 | 9 | 10 | Should display the message Isosceles triangle |
| 14 | Enter the max value for 2 items and max - 1 for any one item | 9 | 10 | 10 | Should display the message Isosceles triangle |
| 15 | Enter the max value for a, b and c | 10 | 10 | 10 | Should display the message Equilateral  triangle |  |  |  |

## Test Case Name :Equivalence class Analysis for triangle problem Experiment Number : 4

**Test Data : Enter the 3 Integer Value( a , b And c ) Classes: { (1,3),(3,5),(5,7)(7,10) }**

**Pre-condition : 1 ≤ a ≤ 10 , 1 ≤ b ≤ 10 and 1 ≤ c ≤ 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 for input data**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Weak Equivalence class Testing** | | | | | | | | |
| **Case Id** | **Description** | **Input Data** | | | **Expected Output** | **Actual Output** | **Status** | **Comments** |
| **a** | **b** | **C** |
| 1 | From class (1,3) | 1 | 1 | 1 | Should display the message Equilateral  triangle |  |  |  |
| 2 | From class(3,5) | 3 | 5 | 3 | Should display the message Isosceles  triangle |  |  |  |
| 3 | From class(3,5) | 3 | 4 | 5 | Should display the message Scalene  triangle |  |  |  |
| 4 | From class(3,5) | 4 | 3 | 5 | Message should be displayed can't form  a triangle |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Weak Robust Equivalence Class Testing** | | | | | | | | |
| 5 | 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 |  |  |  |
| 6 | 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 |  |  |  |
| 7 | 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 |  |  |  |
| 8 | 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 |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | Enter one invalid input and two valid  value for a , b and c | | 5 | 11 | 5 | Should display value of a is not in the  range of permitted values |  |  |  |
| 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.**

## (Dataflow Testing for commission calculation)

#include<stdio.h>

int main()

{

int locks, stocks, barrels, tlocks, tstocks, tbarrels;

float lprice,sprice,bprice,lsales,ssales,bsales,sales,comm;

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) {

printf("enter the number of stocks and barrels\n"); scanf("%d%d",&stocks,&barrels);

tlocks=tlocks+locks;

tstocks=tstocks+stocks;

tbarrels=btarrels+barrels;

printf("\nenter the number of locks and to exit the loop enter -1 for locks\n"); scanf("%d",&locks);

}

printf("\ntotal locks = %d\”,tlocks);

printf(“total stocks =%d\n”,tstocks);

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

lsales = lprice\*tlocks;

ssales=sprice\*tstocks;

bsales=bprice\*tbarrels;

sales=lsales+ssales+bsales;

printf("\nthe total sales=%f\n",sales);

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);

return 0;

}

## Test Case Name : Data Flow Testing for Commission Program Experiment No : 5

**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 | 32,33,37,40 |

**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,40) | ( (31,32,33),40) | <31-32-33-40> | Yes |  |
| ((34 , 35) , 40) | <34-35-40> | Yes |  |
| ((39 , 40 ) | <39 - 40> | Yes |  |

## Program 6 (Binary Search - Path Testing)

**/\* 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; while(low<=high)

{

mid=(low+high)/2; if(key==x[mid])

return mid; if(key<x[mid])

high=mid-1;

else

}

low=mid+1;

return -1;

}

int main()

{

int x[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",&x[i]);

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

succ=binsrc(x,0,n-1,key); if(succ>=0)

printf("Element found in position = %d\n",succ+1);

}

else

else

printf("Element not found \n");

printf("Number of element should be greater than zero\n"); return 0;

}

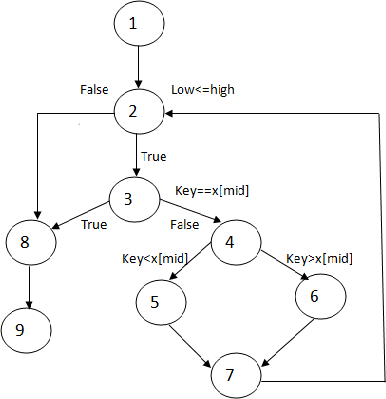
## Binary Search function with line number

int binsrc(int x[],int low,int high,int key)

{

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| while(low<=high) | | |  |  |
| { | | |  |
| mid=(low+high)/2; | | |  |
| if(key==x[mid]) | | |  |
| return mid; | | |  |
| if(key<x[mid]) | | |  |  |
| high=mid-1; | | |  |  |
| else | | |  |  |
|  |  | low=mid+1; |  | |
|  | } |  |  | |
|  | return -1; |  |  | |
| } |  |  |  | |

**Program Graph – for Binary Search**







**Test Cases – Binary Searc**