

**School of Computing Science and Engineering****Software Testing & Quality Assurance****Course Code: E2UC502T****Target Group: BTech V Sem****Practice Problems****Practice Problem 1:****Test Case Name : Boundary Value Analysis for triangle problem**

Design and develop a program in a Java language to solve the triangle problem defined as follows. Accept three integers which are supposed to be the three sides of a triangle and determine the three values represent an equilateral triangle, isosceles triangle, and 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, and execute the test cases and discuss the results.

**Source Code:**

```
import java.util.Scanner;

public class triangle_BVA {
    public static void main(String[] args) {
        int t=0;
        System.out.println("Enter 3 integers which are sides of triangle\n");
        Scanner in = new Scanner(System.in);
        int a = in.nextInt();
        int b = in.nextInt();
        int c = in.nextInt();
        in.close();
    }
}
```

```
if(a<0 || a>10)
{
    System.out.println("a is out of range");
    t=1;
}
if(b<0 || b>10)
{
    System.out.println("b is out of range");
    t=1;
}
if(c<0 || c>10)
{
    System.out.println("c is out of range");
    t=1;
}
if(!(t==1))
{
    if(a<b+c && b<a+c && c<a+b)
    {
        System.out.println("triangle can be formed\n");
        if((a==b)&&(b==c))
            System.out.println("Equilateral triangle");
        else if((a!=b)&&(a!=c)&&(b!=c))
            System.out.println("Scalene triangle");
        else
            System.out.println("Isosceles triangle");
    }
    else
```

```

        System.out.println("triangle cannot be formed");
    }
    else
        return;
    }
}

```

### Sample Output

```

enter 3 integers which are sides of triangle
5
5
1
a=5      b=5      c=1isosceles triangle
enter 3 integers which are sides of triangle
5
1
5
a=5      b=1      c=5isosceles triangle

```

**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 an Equilateral, Isosceles, Scalene triangle or can't form a triangle.

	Min	Min + 1	Normal	Max-1	Max
<b>a</b>	<b>1</b>	<b>2</b>	<b>5</b>	<b>9</b>	<b>10</b>
<b>b</b>	<b>1</b>	<b>2</b>	<b>5</b>	<b>9</b>	<b>10</b>
<b>c</b>	<b>1</b>	<b>2</b>	<b>5</b>	<b>9</b>	<b>10</b>

**Triangle Problem – Boundary Value Test Case for Input Data**

Case ID	Description	Input Data			Expected Output	Actual Output	Status
		a	b	c			
1	Enter the normal value for a, b and c min value for c	5	5	1	Should display the message Isosceles triangle		
2	Enter the normal value for a, b and min + 1 value for c	5	5	2	Should display the message Isosceles triangle		
3	Enter the normal value for a, b and c	5	5	5	Should display the message Equilateral triangle		
4	Enter the normal value for a, b and max – 1 value for c	5	5	9	Should display the message Isosceles triangle		
5	Enter the normal value for a, b and max value for c	5	5	10	Should display the message Not a triangle		
6	Enter the normal value for a, c and min value for b	5	1	5	Should display the message Isosceles triangle		
7	Enter the normal value for a, c and min + 1 for b	5	2	5	Should display the message Isosceles triangle		
8	Enter the normal value for a, c and max – 1 value for b	5	9	5	Should display the message Isosceles triangle		
9	Enter the normal value for a, c and max value for b	5	10	5	Should display the message Not a triangle		
10	Enter the normal value for b, c and min value for a	1	5	5	Should display the message Isosceles triangle		
11	Enter the normal value for b, c and min + 1 value for a	2	5	5	Should display the message Isosceles triangle		
12	Enter the normal value for b, c and max – 1 for a	9	5	5	Should display the message Isosceles triangle		
13	Enter the normal value for b, c and max value for a	10	5	5	Should display the message Not a triangle		
14	Enter the value for a, b, c	2	3	4	Should display the message Scalene triangle		

**Note:**

**After executing the Java program, record the "Actual Output and Status" of your test cases.**

**Practice Problem 2:****Test Case Name : Triangle problem using Decision Table.**

Design and develop a program in a Java Language 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.

**Source Code:**

```
import java.util.Scanner;

public class triangle_DT {
    public static void main(String[] args) {
        System.out.println("\n Enter 3 integers which are sides of triangle\n");
        Scanner in = new Scanner(System.in);
        int a = in.nextInt();
        int b = in.nextInt();
        int c = in.nextInt();
        in.close();

        if(a<b+c && b<a+c && c<a+b)
        {
            System.out.println("triangle can be formed");
            if((a==b) && (b==c))
                System.out.println("equilateral triangle");
            else if((a!=b) && (a!=c) && (b!=c))
                System.out.println("scalene triangle");
            else
                System.out.println("isosceles triangle");
        }
    }
}
```

```
    }  
    else  
        System.out.println("triangle cannot be formed");  
    }  
}
```

### Sample Output

```
Enter 3 integers which are sides of triangle  
  
10  
10  
10  
triangle can be formed  
equilateral triangle  
|
```

<b>Test Case Name</b>	<b>: Decision Table for Triangle Problem</b>
<b>Experiment Number</b>	<b>: 2</b>
<b>Test Data</b>	<b>: Enter the 3 integer value (a, b and c)</b>
<b>Pre-condition</b>	<b>: <math>a &lt; b + c</math>, <math>b &lt; a + c</math> and <math>c &lt; a + b</math></b>
<b>Brief Description</b>	<b>: Check whether given value for an equilateral, isosceles, scalene triangle or cannot for a triangle</b>

**Input Data Decision Table**

Rules		R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
Condition	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 Case for Input Data**

Case ID	Description	Input Data			Expected Output	Actual Output	Status
		a	b	c			
1	Enter the value of a, b and c such that a is less than sum of two sides	20	5	5	Message should be displayed can't form a triangle		
2	Enter the value of a, b and c such that b is 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		
3	Enter the value of a, b and c such that c is 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		
4	Enter the value a, b and c satisfying pre-condition and $a = b$ , $b = c$ and $c = a$	5	5	5	Should display the Equilateral triangle		
5	Enter the value a, b and c satisfying pre-condition and $a = b$ and $b \neq c$	10	10	9	Should display the Isosceles triangle		
6	Enter the value a, b and c satisfying pre-condition and $a \neq b$ , $b \neq c$ and $c \neq a$	5	6	7	Should display the Scalene triangle		

**Note: After executing the Java program, record the "Actual Output and Status" of your test cases.**

**Practice Problem: 3****Test Case Name : Equivalence Class Analysis for triangle problem**

Design and develop a program in a Java language 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, execute the test cases and discuss the results.

**Source Code**

```
import java.util.Scanner;

public class triangle_EC{
    public static void main(String[] args) {

        int t=0;
        System.out.println("Enter 3 integers which are sides of triangle\n");
        Scanner in = new Scanner(System.in);
        int a = in.nextInt();
        int b = in.nextInt();
        int c = in.nextInt();
        in.close();

        if(a<0 || a>10)
        {
            System.out.println("a is out of range");
            t=1;
        }
        if(b<0 || b>10)
        {
            System.out.println("b is out of range");
```



```
t=1;
}
if(c<0 || c>10)
{
    System.out.println("c is out of range");
    t=1;
}
if(!(t==1))
{
    if(a<b+c && b<a+c && c<a+b)
    {
        System.out.println("triangle can be formed\n");
        if((a==b)&&(b==c))
            System.out.println("Equilateral triangle");
        else if((a!=b)&&(a!=c)&&(b!=c))
            System.out.println("Scalene triangle");
        else
            System.out.println("Isosceles triangle");
    }
    else
        System.out.println("triangle cannot be formed");
}
else
    return;
}
}
```

**Sample Output**

```
Enter 3 integers which are sides of triangle

11
11
11
a is out of range
b is out of range
c is out of range
```

<b>Test Case Name</b>	<b>: Equivalence Class Analysis for Triangle Problem</b>
<b>Experiment Number</b>	<b>: 3</b>
<b>Test Data</b>	<b>: Enter the 3 Integer Value (a, b and c)</b>
<b>Pre-condition</b>	<b>: <math>1 \leq a \leq 10</math>, <math>1 \leq b \leq 10</math>, and <math>1 \leq c \leq 10</math> and <math>a &lt; b + c</math>, <math>b &lt; a + c</math>, and <math>c &lt; a + b</math></b>
<b>Brief Description</b>	<b>: Check whether given value for an Equilateral, Isosceles, and Scalene triangle or cannot form a triangle</b>
<b>Valid Classes</b>	<b>: <math>1 \leq a \leq 10</math>, <math>1 \leq b \leq 10</math>, <math>1 \leq c \leq 10</math></b>
<b>Invalid Classes</b>	<b>: <math>a &lt; 1</math> &amp; <math>a &gt; 10</math>, <math>b &lt; 1</math> &amp; <math>b &gt; 10</math>, <math>c &lt; 1</math> &amp; <math>c &gt; 10</math></b>

**Triangle Problem: Equivalence Class Test Cases for Input Data**

<b>Weak Equivalence Class Testing</b>							
<b>Case ID</b>	<b>Description</b>	<b>Input Data</b>			<b>Expected Output</b>	<b>Actual Output</b>	<b>Status</b>
		<b>a</b>	<b>b</b>	<b>c</b>			
1	Enter the min value for a, b and c	5	5	5	Should display the message Equilateral triangle		
2	Enter the min value for a, b and c	2	2	3	Should display the message Isosceles triangle		
3	Enter the min value for a, b and c	3	4	5	Should display the message Scalene triangle		
4	Enter the min value for a, b and c	4	1	2	Message should be displayed cannot form a triangle		

<b>Weak Robust Equivalence Class Testing</b>							
<b>Case ID</b>	<b>Description</b>	<b>Input Data</b>			<b>Expected Output</b>	<b>Actual Output</b>	<b>Status</b>
		<b>a</b>	<b>b</b>	<b>c</b>			
1	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		
2	Enter one invalid input and two valid value for a, b and c	5	-1	5	Should display value of b is not in the range of permitted values		
3	Enter one invalid input and two valid value for a, b and c	5	5	-1	Should display value of c is not in the range of permitted values		
4	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		
5	Enter one invalid input and two valid value for a, b and c	5	11	5	Should display value of b is not in the range of permitted values		
6	Enter one invalid input and two valid value for a, b and c	5	5	11	Should display value of c is not in the range of permitted values		

<b>Strong Robust Equivalence Class Testing</b>							
<b>Case ID</b>	<b>Description</b>	<b>Input Data</b>			<b>Expected Output</b>	<b>Actual Output</b>	<b>Status</b>
		<b>a</b>	<b>b</b>	<b>c</b>			
1	Enter one invalid input and two value value for a, b and c	-1	5	5	Should display value of a is not in the range of permitted values		
2	Enter one invalid input and two value value for a, b and c	5	-1	5	Should display value of b is not in the range of permitted values		
3	Enter one invalid input and two value value for a, b and c	5	5	-1	Should display value of c is not in the range of permitted values		
4	Enter two invalid input and one 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		
5	Enter two invalid input and one 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 b is not in the range of permitted values		
6	Enter two invalid input and one 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		

7	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		

**Note: After executing the Java program, record the "Actual Output and Status" of your test cases.**

**Practice Problem: 4****Test Case Name : Equivalence class test cases for Next date**

Design, develop, code and run the program in Java language to implement the NextDate function. Analyse it from the perspective of equivalence class analysis. Derive different test cases, execute these test cases and discuss the test results.

**Source code:****import java.util.Scanner;****public class date\_EC {****public static void main(String[] args) {****int day,month,year,tomm\_day = 0,tomm\_month = 0, tomm\_year = 0;****System.out.println("\n Enter the today's date in the form of dd mm  
yyyy\n");****Scanner in = new Scanner(System.in);****day = in.nextInt();****month = in.nextInt();****year = in.nextInt();****in.close();****tomm\_month=month;****tomm\_year=year;****if(day<1 || day>31)****System.out.println("Value of day, not in the range 1.....31\n");**

```
if(month<1 || month>12)
    System.out.println("Value of month, not in the range
1.....12\n");

if(year<1812 || year>2013)
    System.out.println("Value of year, not in the range
1812.....2013\n");

else if(check(day,month))
    System.out.println("Value of day, not in the range day<=30");

if(month==2)
{
    if(isleap(year) && day>20)
        System.out.println("invalid date input for leap year");
}

else if(!(isleap(year))&& day>28)
    System.out.println("Invalid date input for not a leap
year");

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;

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;
    tomm_year=year+1;
}
break;
```



**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;**

**}**

```
System.out.println("Next day is: " + tomm_day + " " + tomm_month + " " +  
tomm_year);  
}
```

**private static boolean isleap(int year) {**

**//To do Auto generated method stub**

**if((year%4==0 && year%100!=0) || year%400==0)**

**return true;**

**else**

**return false;**

**}**

**private static boolean check(int day, int month) {**

**//To do Auto-generated method stub**

**if((month==4||month==6||month==9||month==11) && day==31)**

**return true;**

**else**

**return false;**

**}**

**}**

**Sample Output:**

```
Enter the today's date in the form of dd mm yyyy

10
10
2000
Next day is: 11 10 2000
|
```

**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 AND 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**

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

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

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

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

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

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

**Next Date Output Equivalence Class Testing**

<b>Weak and Strong Normal Equivalence Class</b>											
<b>Case ID</b>	<b>Description</b>	<b>Input Data</b>			<b>Expected Output</b>			<b>Actual Output</b>			<b>Status</b>
		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				

**Note:**

- **WN → Weak Normal**
- **SN → Strong Normal**
- **WR → Weak Robustness**
- **SR → Strong Robustness**

<b>Weak Robustness Equivalence Class</b>											
<b>Case ID</b>	<b>Description</b>	<b>Input Data</b>			<b>Expected Output</b>			<b>Actual Output</b>			<b>Status</b>
		Month	Day	Year	Month	Day	Year	Month	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 on the range 1...12						
WR3	Enter the M3, D1 and Y1 cases	13	15	1912	Should display the message value of the month not on the range 1...12						
WR4	Enter the M1, D2 and Y1 cases	6	-1	1912	Should display the message value of the month not on the range 1...31						
WR5	Enter the M1, D3 and Y1 cases	6	32	1912	Should display the message value of the month not on the range 1...31						
WR6	Enter the M1, D1 and Y2 cases	6	15	1811	Should display the message value of the month not on the range 1812....2013						
WR7	Enter the M1, D1 and Y3 cases	6	15	2014	Should display the message value of the month not on the range 1812....2013						

Strong Robustness Equivalence Class											
Case ID	Description	Input Data			Expected Output			Actual Output			Status
		Month	Day	Year	Month	Day	Year	Month	Day	Year	
SR1	Enter the M2, D1 and Y1 cases	-1	15	1912	Should display the message value of the month not on the range 1...12						
SR2	Enter the M1, D2 and Y1 cases	6	-1	1912	Should display the message value of the month not on the range 1...31						
SR3	Enter the M1, D1 and Y2 cases	6	15	1811	Should display the message value of the month not on the range 1812....2013						
SR4	Enter the M2, D2 and Y1 cases	-1	-1	1912	(i) Should display the message value of the month not on the range 1...12						
					(ii) Should display the message value of the month not on the range 1...31						
SR5	Enter the M1, D2 and Y2 cases	6	-1	1811	(i) Should display the message value of the month not on the range 1...31						
					(ii) Should display the message value of the month not on the range 1...12						
SR6	Enter the M2, D1 and Y2 cases	-1	15	1811	(i) Should display the message value of the month not on the range 1....12						
					(ii) Should display the message value of the month not on the range 1812....2013						

SR7	Enter the M2, D2 and Y2 cases	-1	-1	1811	(i) Should display the message value of the month not on the range 1.....12				
					(ii) Should display the message value of the month not on the range 1.....31				
					(iii) Should display the message value of the month not on the range 1812....2013				

Some addition Equivalence Boundary Checking											
Case ID	Description	Input Data			Expected Output			Actual Output			Status
		Day	Month	Year	Day	Month	Year	Month	Day	Year	
1	Enter the D1, M1 and Y1 valid cases	31	12	1811	Should display the message value of the month not on the range 1812....2013						
2	Enter the D1, M1 and Y2	31	12	2012	1	1	2013				
3	Enter the D1, M1 and Y3 valid cases	31	12	2013	Should display the message value of the month not on the range 2013						

**Note: After executing the Java program, record the "Actual Output and Status" of your test cases.**

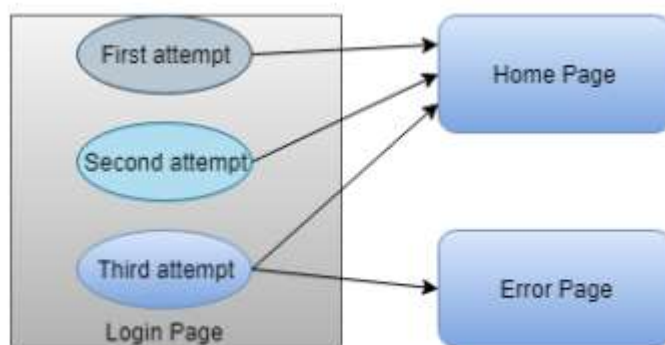
**Practice Problem: 5**

We all use the ATMs, when we withdraw money from it, it displays account details at last. Now we again do another transaction, then it again displays account details, but the details displayed after the second transaction are different from the first transaction, but both details are displayed by using the same function of the ATM. So the same function was used here but each time the output was different, this is called state transition. In the case of testing of a software application, this method tests whether the function is following state transition specifications on entering different inputs.



The image shows a login form titled "Login Here" on a purple background. It contains two input fields: "Email" with a placeholder "Enter Your Email" and "Password" with a placeholder "10 characters". Below the fields are two buttons: "Back" and "Login".

**Let see the diagram to solve this problem**



STATE	LOGIN	VALIDATION	REDIRECTED
S1	First Attempt	Invalid	S2
S2	Second Attempt	Invalid	S3
S3	Third Attempt	Invalid	S5
S4	Home Page		
S5	Error Page		

**Test case 2: if the third attempt is valid, then it will be directed to the homepage.**

STATE	LOGIN	VALIDATION	REDIRECTED
S1	First Attempt	Invalid	S2
S2	Second Attempt	Invalid	S3
S3	Third Attempt	Valid	S4
S4	Home Page		
S5	Error Page		

**Practice Problem: 6****Test Case Name : Boundary Value Analysis for Age program.**

Consider a system that accepts ages from 18 to 56.

<b>Boundary Value Analysis(Age accepts 18 to 56)</b>		
<b>Invalid (min-1)</b>	<b>Valid (min, min + 1, nominal, max – 1, max)</b>	<b>Invalid (max + 1)</b>
<b>17</b>	<b>18, 19, 37, 55, 56</b>	<b>57</b>

**Valid Test cases:** Valid test cases for the above can be any value entered greater than 17 and less than 57.

- ✓ Enter the value- 18.
- ✓ Enter the value- 19.
- ✓ Enter the value- 37.
- ✓ Enter the value- 55.
- ✓ Enter the value- 56.

**Invalid Test cases:** When any value less than 18 and greater than 56 is entered.

- ✓ Enter the value- 17.
- ✓ Enter the value- 57.



**Practice Problem: 7**

**Test Case Name : Railway Registration.**

**Test cases for Railway Registration:**

1. Check whether there is a train for your destination.
2. Check the availability of seats on the required date.
3. Check whether the available seats for required coach that is A/c or Sleeper coaches.

**Test Scenario: Train search with all route stations**

1. Enter source and destination codes or source and destination corresponding trains are displaying or not.
2. Enter train number and date source and destination with all stations are displaying or not.

**Test Scenario: check availability of seats for required date**

1. Search seats for Ac, sleeper, seat.
2. Check upper, middle, lower births.
3. Ticket fare for corresponding criteria
4. Booking: Book the tickets with corresponding criteria by cash or card any concession for senior citizen if applicable.

**Test Scenario: Print the ticket same for cancelation of ticket**

1. Check whether you have money.
2. Check the train is available & you have to go to the same destination.
3. Check working hours of booking office.

Step 1: Verify there is a train available to the destination.

Step 2: Verify the seats are available on the particular date.

Step 3: Verify the seats available of the coaches like sitting or berth and check the fare of the particular system.

**Practice Problem: 8****Test Case Name : Boundary Value Check for  $a^b$** 

Program computes  $a^b$  where  $a$  lies in range  $[1,10]$  and  $b$  lies in range  $[1,5]$ . Design the test cases for Boundary Value Check. Also calculate the number of test cases.

**Source Code**

```
class Main {  
    public static void main(String[] args) {  
        Scanner sc= new Scanner(System.in);  
        int a= sc.nextInt();  
        int b= sc.nextInt();  
        long result = 1;  
        while (b != 0) {  
            result *= a;  
            --b;  
        }  
        System.out.println("Answer = " + result);  
    }  
}
```

	FOR a	For b
Max	10	5
Min	1	1
Max-1	9	4
Min+1	2	2
Nominal Value	5	3

No. of test cases:  $4n+1 = 4*2+1 = 9$  test cases

**Design of test cases for Boundary Value Check:**

Test case	a	b	$a^b$
1	5	5	3125
2	5	1	5
3	5	4	625
4	5	2	25
5	10	3	100
6	1	3	1
7	9	3	27
8	2	3	8
9	5	3	125

## Practice Problem: 9

Assume that, age is a variable of any function, and its minimum value is 18 and the maximum value is 30, both 18 and 30 will be considered as boundary values are those that contain the upper and lower limit of a variable. For example There is 18 and 30 are the boundary values that's why tester pays more attention to these values, but this doesn't mean that the middle values like 19, 20, 21, 27, 29 are ignored. Test cases are developed for each and every value of the range.

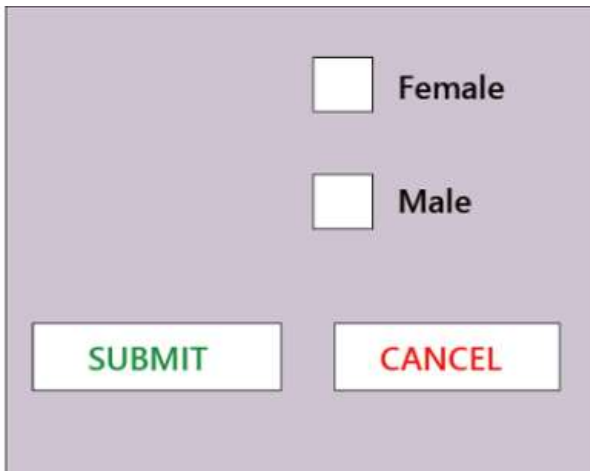



Invalid test cases	Valid test cases	Invalid test cases
11, 13, 14, 15, 16, 17	18, 19, 24, 27, 28, 30	31, 32, 36, 37, 38, 39

**Practice Problem 10:**

If the requirement ID Boolean (true/false), then derive the test case for both true/false values.

The Boolean value can be true and false for the radio button, checkboxes. For example



The form is a light purple rectangle. It contains two radio buttons stacked vertically. The top one is labeled 'Female' and the bottom one is labeled 'Male'. Below these are two rectangular buttons: 'SUBMIT' in green text and 'CANCEL' in red text.

Serial no	Description	Input	Expected	Note
1	Select valid	NA	True	---
2	Select invalid	NA	False	Values can be change based according to the requirement.
3	Do not select	NA	Do not select anything, error message should be displayed	We cannot go for next question
4	Select both	NA	We can select any radio button	Only one radio button can be selected at a time.

\*\*\*\*\*