EXPERIMENT 6

- ASHISH KUMAR
- 2K18/SE/041

AIM:- Write a program to generate test cases(for types of triangle) using Decision Table Testing.

THEORY:- Decision tables are used in many engineering disciplines to represent complex logical relationships. An output may be dependent on many input conditions and decision tables give a pictorial view of various combinations of input conditions. There are four portions of the decision table. The four parts of the decision table are given as:

1. Condition Stubs: All the conditions are represented in this upper left section of the decision table. These conditions are used to determine a particular action or set of actions.
2. Action Stubs: All possible actions are listed in this lower left portion of the decision table.
3. Condition Entries: In the condition entries portion of the decision table, we have a number of columns and each column represents a rule. Values entered in this upper right portion of the table are known as inputs.
4. Action Entries: Each entry in the action entries portion has some associated action or set of actions in this lower right portion of the table. These values are known as outputs and are dependent upon the functionality of the program.

CODE:-

```
#include <bits/stdc++.h>
#include<iostream>
#include<cstring>
using namespace std;


int printArr[7][3];
int a, b, c;
string OutputType(int a, int b, int c, int arr[3][3]) {
 string ans = "Not a Triangle";
 if (a < arr[0][0] || a > arr[0][1] || b < arr[1][0] || b > arr[1][1] || c < arr[2][0] || c > arr[2][1]) {
    ans = "Input values are out of range";
 }
 else if (a < b + c && b < a + c && c < a + b) {
       if (a == b && b == c) {
          ans = "Equilateral Triangle";
                    }
                    else if (a == b || b == c || a == c) {
                       ans = "Isosceles Triangle";
                    }
                    else {
                        ans = "Scalene Triangle";
                    }
                }
```

```
        return ans;
}

string DecisionTable() {
 cout<<"\nDECISION TABLE FOR TRIANGLE CLASSIFICATION PROBLEM"<<endl;
 string str = " ---------------------------------------------------------------\n";
 str = str + " Decisions          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11|\n";
 str = str + " ---------------------------------------------------------------\n";
 str = str + " C1: a < b + c?     | F | T | T | T | T | T | T | T | T | T |\n";
 str = str + " C2: b < a + c?     | - | F | T | T | T | T | T | T | T | T |\n";
 str = str + " C3: c < a + b?     | - | - | F | T | T | T | T | T | T | T |\n";
 str = str + " C4: a = b ?        | - | - | - | T | F | T | F | T | T | F | F |\n";
 str = str + " C5: a = c ?        | - | - | - | T | F | F | T | T | F | T | F |\n";
 str = str + " C6: b = c ?        | - | - | - | T | T | F | F | F | T | T | F |\n";
 str = str + " ---------------------------------------------------------------\n";
 str = str + " Rule count         |32 |16 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |\n";
 str = str + " ---------------------------------------------------------------\n";
 str = str + " A1: Not a triangle | X | X | X |  |  |  |  |  |  |  |  |\n";
 str = str + " A2: Scalene        |  |  |  |  |  |  |  |  |  |  | X |\n";
 str = str + " A3: Isosceles      |  |  |  |  | X | X | X |  |  |  |  |\n";
 str = str + " A4: Equilateral    |  |  |  | X |  |  |  |  |  |  |  |\n";
 str = str + " A5: Impossible     |  |  |  |  |  |  |  | X | X | X |  |\n";
 return str;
}
```

```
int notTriangle(int index, int arr[3][3]) {
 a = arr[0][1];
 b = arr[1][2];
 c = arr[2][2];
  while (a < (b + c)) {
    b = b / 2;
    c = c / 2;
 }
 printArr[0][0] = a;
 printArr[0][1] = b;
 printArr[0][2] = c;
 index++;
  a = arr[0][2];
 b = arr[1][1];
 c = arr[2][2];
 while (b < (a + c)) {
 a = a / 2;
 c = c / 2;
 }
 printArr[1][0] = a;
 printArr[1][1] = b;
 printArr[1][2] = c;
 index++;
 a = arr[0][2];
 b = arr[1][2];
```

```
 c = arr[2][1];
 while (c < (a + b)) {
 a = a / 2;
 b = b / 2;
 }
 printArr[2][0] = a;
 printArr[2][1] = b;
 printArr[2][2] = c;
 index++;
 return index;
}
int Equilateral(int index, int arr[3][3]) {
 a = arr[0][3];
 b = arr[1][3];
 c = arr[2][3];
 int minMax = arr[0][0], maxMin = arr[0][1], nominalEq;
 for (int i = 0; i < 3; i++) {
 if (minMax > arr[i][0])
 minMax = arr[i][0];
 if (maxMin > arr[i][1])
 maxMin = arr[i][1];
 }
 nominalEq = (minMax + maxMin) / 2;
 printArr[index][0] = nominalEq;
 printArr[index][1] = nominalEq;
```

```
 printArr[index][2] = nominalEq;
 index++;
 return index;
}

int Isosceles(int index, int arr[3][3]) {
 a = arr[0][3];
 b = arr[1][3];
 c = arr[2][3];
 int minMax = arr[0][0], maxMin = arr[0][1], nominalEq;
 for (int i = 0; i < 3; i++) {
 if (minMax > arr[i][0])
 minMax = arr[i][0];
 if (maxMin > arr[i][1])
 maxMin = arr[i][1];
 }
 nominalEq = (minMax + maxMin) / 2;
 printArr[index][0] = arr[0][0];
 printArr[index][1] = nominalEq;
 printArr[index][2] = nominalEq;
 index++;
 printArr[index][0] = nominalEq;
 printArr[index][1] = arr[1][0];
 printArr[index][2] = nominalEq;
 index++;
```

```
 printArr[index][0] = nominalEq;
 printArr[index][1] = nominalEq;
 printArr[index][2] = arr[2][0];
 index++;
 return index;
}

int impossible(int index) {
 for (int i = 0; i < 3; i++) {
 cout << index + i << "\t?\t?\t?\tImpossible" << endl;
}
 index = index + 3;
 return index;
}

void printDTTestCases(int arr[3][3]) {
 cout << "S.No\tA\tB\tC\tExpected Output" << endl;
 cout<<"-----------------------------------------------------------\n";
 int index = 0;
 index = notTriangle(index, arr);
 index = Equilateral(index, arr);
 index = Isosceles(index, arr);
 index = 1;
 for (int i = 0; i < 7; i++) {
 a = printArr[i][0];
```

```
 b = printArr[i][1];
 c = printArr[i][2];
 cout << index << "\t" << a << "\t" << b << "\t" << c << "\t" << OutputType(a, b, c, arr) << endl;
 index++;
 }
 index = impossible(index);
 cout << index << "\t" << arr[0][1] - 1 << "\t" << arr[1][2] << "\t" << arr[2][2] + 2 << "\t" << OutputType(arr[0][1] - 1, arr[1][2], arr[2][2] + 2, arr) << endl;
}

int main() {
 int arr[3][3]; //0-min, 1- max, 2- nominal
  for (int i = 0; i < 3; i++) {
 cout << "Enter min & max value of vertex " << i + 1 << ": ";
 cin >> arr[i][0] >> arr[i][1];
 arr[i][2] = (arr[i][0] + arr[i][1]) / 2;
 arr[i][3] = arr[i][1] - arr[i][0];
 }
 cout << DecisionTable() << endl;
 cout<<"TEST CASES..\n"<<endl;
 printDTTestCases(arr);
 cout<<"\n>>Total No. of Test Cases = 11";
 return 0;
}
```

OUTPUT:-

```
C:\Users\Ashish\Desktop\decisiontable.exe
Enter min & max value of vertex 1: 1 10
Enter min & max value of vertex 2: 1 10
Enter min & max value of vertex 3: 1 10

DECISION TABLE FOR TRIANGLE CLASSIFICATION PROBLEM
 ---------------------------------------------------------------
 Decisions          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10| 11|
 ---------------------------------------------------------------
 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 | F | T | F | T | T | F | F |
 C5: a = c ?        | - | - | - | T | F | F | T | T | F | T | F |
 C6: b = c ?        | - | - | - | T | T | F | F | F | T | T | F |
 ---------------------------------------------------------------
 Rule count         |32 |16 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
 ---------------------------------------------------------------
 A1: Not a triangle | X | X | X |   |   |   |   |   |   |   |   |
 A2: Scalene        |   |   |   |   |   |   |   |   |   |   | X |
 A3: Isosceles      |   |   |   |   | X | X | X |   |   |   |   |
 A4: Equilateral    |   |   |   | X |   |   |   |   |   |   |   |
 A5: Impossible     |   |   |   |   |   |   |   | X | X | X |   |

TEST CASES..

S.No    A       B       C       Expected Output
---------------------------------------------------------------
1       10      5       5       Not a Triangle
2       5       10      5       Not a Triangle
3       5       5       10      Not a Triangle
4       5       5       5       Equilateral Triangle
5       1       5       5       Isosceles Triangle
6       5       1       5       Isosceles Triangle
7       5       5       1       Isosceles Triangle
8       ?       ?       ?       Impossible
9       ?       ?       ?       Impossible
10      ?       ?       ?       Impossible
11      9       5       7       Scalene Triangle

>>Total No. of Test Cases = 11
--------------------------------
Process exited after 4.32 seconds with return value 0
Press any key to continue . . .
```