

EXPERIMENT 10

- ASHISH KUMAR

- 2K18/SE/041

AIM:- Write a program to generate test cases using cause effect graph.

THEORY:- Cause Effect Graphing based technique is a technique in which a graph is used to represent the situations of combinations of input conditions. The graph is then converted to a decision table to obtain the test cases. Cause-effect graphing technique is used because boundary value analysis and equivalence class partitioning methods do not consider the combinations of input conditions. But since there may be some critical behaviour to be tested when some combinations of input conditions are considered, that is why cause-effect graphing technique is used.

CODE:-

```
#include<iostream>

#include<conio.h>

#include<bits/stdc++.h>

using namespace std;

int triangle(int a[3], int min=1, int max=100)
{
    for(int i=0;i<3;i++)
        if(a[i]<min || a[i]>max)
            return -1;
```

```

    if(a[0]+a[1]>a[2] && a[1]+a[2]>a[0] && a[2]+a[0]>a[1])
    {
        if(a[0]==a[1] && a[1]==a[2] && a[2]==a[0])
            return 1;
        else if(a[0]==a[1] || a[1]==a[2] || a[2]==a[0])
            return 2;
        else
            return 3;
    }
    else
        return 0;
}

```

```

string cause[6];

```

```

string effect[5];

```

```

char causeitionEntries[6][11];

```

```

char act[6][11];

```

```

void generateCauseAndEffect()

```

```

{
    cause[0] = "c1: a<b+c?";
    cause[1] = "c2: b<c+a?";
    cause[2] = "c3: c<a+b?";
    cause[3] = "c4: a^2=b^2+c^2";
    cause[4] = "c5: a^2>b^2+c^2";
    cause[5] = "c6: a^2<b^2+c^2";
    effect[0] = "e1: Invalid Triangle";
}

```

```

effect[1] = "e2: Right Angle Triangle";
effect[2] = "e3: Obtuse Angled Triangle";
effect[3] = "e4: Acute Angled Triangle";
effect[4] = "e5: Impossible";

//generating causeition Entries for  $a < b + c$ ?

int i=0;

causeitionEntries[0][i]='0';

for(i=1;i<11;i++)

    causeitionEntries[0][i]='1';


//for  $b < c + a$ ?

i=0; causeitionEntries[1][i]='X';

i=1; causeitionEntries[1][i]='0';

for(i=2;i<11;i++)

    causeitionEntries[1][i]='1';


//for  $c < a + b$ ?

i=0; causeitionEntries[2][i]='X';

i=1; causeitionEntries[2][i]='X';

i=2; causeitionEntries[2][i]='0';

for(i=3;i<11;i++)

    causeitionEntries[2][i]='1';


//for  $a^2 = b^2 + c^2$ 

for(i=0;i<3;i++)

```

```

        causeitionEntries[3][i]='X';

        causeitionEntries[3][10]='0';
for(i=3;i<10;i++)
if(i<7)
        causeitionEntries[3][i]='1';
else
        causeitionEntries[3][i]='0';

//for  $a^2 > b^2 + c^2$ ?
for(i=0;i<3;i++)
        causeitionEntries[4][i]='X';
for(i=3;i<11;i++)
        if(i==3||i==4|| (i>6&& i<9))
                causeitionEntries[4][i]='1';
        else
                causeitionEntries[4][i]='0';
causeitionEntries[4][9]='0';

//for  $a^2 > b^2 + c^2$ 
for(i=0;i<3;i++)
        causeitionEntries[5][i]='X';
for(i=3;i<11;i++)
        if(i==4||i==6)
                causeitionEntries[5][i]='0';

```

else

```
causeitionEntries[5][i]='1';
```

```
causeitionEntries[5][8]='0';
```

```
//generating effect Entries
```

```
for(i=0;i<5;i++)
```

```
for(int j=0;j<11;j++)
```

```
    act[i][j]=' ';
```

```
    act[0][0]='1';
```

```
    act[0][1]='1';
```

```
    act[0][2]='1';
```

```
    act[1][6]='1';
```

```
    act[2][8]='1';
```

```
    act[3][9]='1';
```

```
    act[4][3]='1';
```

```
    act[4][4]='1';
```

```
    act[4][5]='1';
```

```
    act[4][7]='1';
```

```
    act[4][10]='1';
```

```
    cout<<"\nDecision Table...\n";
```

```
    for(i=0;i<77;i++) cout<<"-"; cout<<"\n";
```

```
    for(i=0;i<6;i++)
```

```
    {
```

```
        cout<<"| "<<setw(30)<<left<<cause[i];
```

```
        cout<<"| ";
```

```

        for(int j=0;j<11;j++)
        {
            cout<<causeitionEntries[i][j]<<" | ";
        }
        cout<<endl;
    }
    for(i=0;i<77;i++) cout<<"-"; cout<<"\n";
    for(i=0;i<5;i++)
    {
        cout<<"| "<<setw(30)<<left<<effect[i];
        cout<<"| ";
        for(int j=0;j<11;j++)
        {
            cout<<act[i][j]<<" | ";
        }
        cout<<endl;
    }
    for(i=0;i<77;i++) cout<<"-"; cout<<"\n";
}

void causeEffectTesting()
{
    int expOut[11], testCases[11][3], i=0, x=0, y=0, z=0;
    generateCauseAndEffect();
}

```

```

for(i=0;i<11;i++)
{
    switch(i)
    {
        case 0: x=90; y=60; z=20;
            break;
        case 1: x=60; y=90; z=20;
            break;
        case 2: x=20; y=60; z=90;
            break;
        case 3: x=50; y=40; z=30;
            break;
        case 4: x=60; y=40; z=30;
            break;
        case 5: x=40; y=60; z=30;
            break;
    }
    testCases[i][0]=x;testCases[i][1]=y;testCases[i][2]=z;
}

cout<<"\nTest Cases using cause effect graph are: \n";
cout<<"\n"; for(int i=0;i<81;i++) cout<<"-";
cout<<"\n\t\t Inputs\t\t\t\t\t";
cout<<setw(30)<<left<<"\tExpected Output \t"<<endl;
cout<<"\tA\t\tB\t\tC \t\t\t\t\t\n";

for(int i=0;i<81;i++) cout<<"-"; cout<<"\n";

```

```

for(int i=0;i<6;i++)
{
    for(int j=0;j<3;j++)
    {
        if(j==0) cout<<"|";

        cout<<"\t"<<testCases[i][j]<<"\t|";

    }
    if(i<3){
        cout<<setw(30)<<left<<effect[0]<<endl;
    }
    else{
        cout<<setw(30)<<left<<effect[i-2]<<endl;
    }
}

for(int i=0;i<81;i++) cout<<"-";

cout<<"\n Total No. of Test Cases = 6";

}

int main()
{
    causeEffectTesting();

    return 0;
}

```


OUTPUT:-

```
C:\Users\Ashish\Desktop\causeEffectgraph.exe

Decision Table...
-----
| c1: a<b+c? | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| c2: b<c+a? | X | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| c3: c<a+b? | X | X | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| c4: a^2=b^2+c^2 | X | X | X | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| c5: a^2>b^2+c^2 | X | X | X | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| c6: a^2<b^2+c^2 | X | X | X | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
-----
| e1: Invalid Triangle | 1 | 1 | 1 | | | | | 1 | | | | |
| e2: Right Angle Triangle | | | | | | | | | 1 | | |
| e3: Obtuse Angled Triangle | | | | | | | | | | 1 | |
| e4: Acute Angled Triangle | | | | | 1 | 1 | 1 | | | 1 | 1 |
| e5: Impossible | | | | 1 | 1 | 1 | | | 1 | | | 1 |
-----

Test Cases using cause effect graph are:
-----
| Inputs | Expected Output |
| A | B | C | |
|---|---|---|---|
| 90 | 60 | 20 | e1: Invalid Triangle |
| 60 | 90 | 20 | e1: Invalid Triangle |
| 20 | 60 | 90 | e1: Invalid Triangle |
| 50 | 40 | 30 | e2: Right Angle Triangle |
| 60 | 40 | 30 | e3: Obtuse Angled Triangle |
| 40 | 60 | 30 | e4: Acute Angled Triangle |
|-----|-----|

Total No. of Test Cases = 6
-----

Process exited after 0.3026 seconds with return value 0
Press any key to continue . . .
```

