**Test Cases for Tip Calculator:-**

* **To check whether it is compatible on all the browsers or not?**
* **To check if JavaScript is enabled in the browser.**
* **To validate the input containers whether they are working properly or not ?**
* **To validate that input type in the input containers are “Numbers”.**
* **To check whether all the arithmetic operations necessary for calculations are working properly or not by entering the values in the containers and checking the result accordingly.**
* **To check that BODMAS rule is followed in arithmetic operations.**
* **To check whether the input fields are clearable (Editable in order to calculate non-identical values).**
* **To validate that the results addressed in the out-put container is clearly visible or not?**
* **To attest that proper CSS is inserted which facilitate us with proper visibility.**
* **To check of when no number entered in the input fields no output should be displayed(NaN).**
* **To insure that a message box should be present for displaying Payment-Methods.**
* **To check that messages are displayed in red color in order to make it visible different from others.**
* **To check of a default value of a Tip (quite general type value).**
* **To check value of (No. of people) is not zero(0).**
* **To validate that percentage of tip cannot be zero.**
* **To insure that maximum value of (No. of people) should not be greater than (>) 50 as it is practically not possible.**
* **To check that input fields are restricted with decimals in order to avoid confusion.**
* **To validate of a feature that if the amount(input) is too high a alert should be displayed which provides convenience in Error detection(if some of the number is tap multiple times leading to increase the value too high).**
* **To validate of a Alert feature to ensure that tip is not 100% because that clearly makes no sense.**

Functional Testing:-

Boundary Value Testing:-

Boundary testing is the process of testing between extreme ends or boundaries values partitions. **2-Major inputs are Tip percentage and the Bill Amount.**

Range for Tip [0-100]:- Tip value can’t exceeds by 100 (0<=Tip<=100).

Range for Bill Amount [50-10000]:- Bill amount can’t be less than 50 and not exceeds 10000 (50<=Amount<=10000).

1. Minimum
2. Just above the minimum
3. A nominal value(Average)
4. Just below the maximum
5. Maximum

Test Cases formed are 4n+1: n is equal to number of testing values here value of n is 2. So the total test cases are (4n+1 = 4\*2+1 => 9).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Program: Tip calculator | | | |  |  |
| Testing Technique: | | | Boundary Value Testing | | |
|  |  |  | Expected Output | |  |
| Test ID | Inputs | | Outputs | | Remark |
| tipPercent | bill | totalAmount | totalTip |
| BV\_TC\_1 | 1 | 5050 | 5100.5 | 50.5 | Pass |
| BV\_TC\_2 | 2 | 5050 | 5151 | 101 | Pass |
| BV\_TC\_3 | 50 | 5050 | 7575 | 2525 | Pass |
| BV\_TC\_4 | 99 | 5050 | 10049.5 | 4999.5 | Pass |
| BV\_TC\_5 | 100 | 5050 | 10100 | 5050 | Pass |
| BV\_TC\_6 | 50 | 50 | 75 | 25 | Pass |
| BV\_TC\_7 | 50 | 51 | 75.5 | 25.5 | Pass |
| BV\_TC\_8 | 50 | 9999 | 14998.5 | 4999.5 | Pass |
| BV\_TC\_9 | 50 | 10000 | 15000 | 5000 | Pass |

**Black-Box Testing-**

**Cause Effect Graphing based technique**:-

**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.

**Basic Notations used in Cause-effect graph:**

Here **c** represents **cause** and **e** represents **effect**.

The following notations are always **used between a cause and an effect**:

1. **Identity Function:** if c is 1, then e is 1. Else e is 0.

**IDENTITY**

1. **NOT Function:** if c is 1, then e is 0. Else e is 1.

**NOT**

1. **OR Function:** if c1 or c2 or c3 is 1, then e is 1. Else e is 0.

**OR**

V

1. **AND Function:** if both c1 and c2 and c3 is 1, then e is 1. Else e is 0.

**AND**

^

To represent some impossible combinations of causes or impossible combinations of effects, constraints are used. The following **constraints** are used in cause-effect graphs:

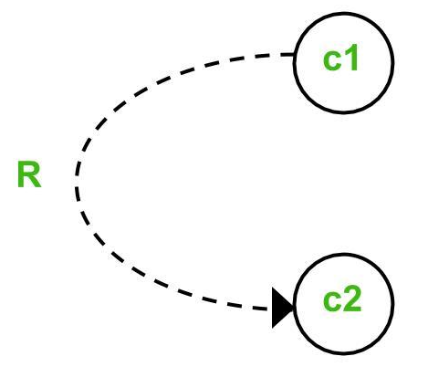
1. **Exclusive constraint** or **E-constraint:** This constraint exists between causes. It states that either c1 or c2 can be 1, i.e., c1 and c2 cannot be 1 simultaneously. **Inclusive constraint**or **I-constraint:** This constraint exists between causes. It states that at least one of c1, c2 and c3 must always be 1, i.e., c1, c2 and c3 cannot be 0 simultaneously.

E

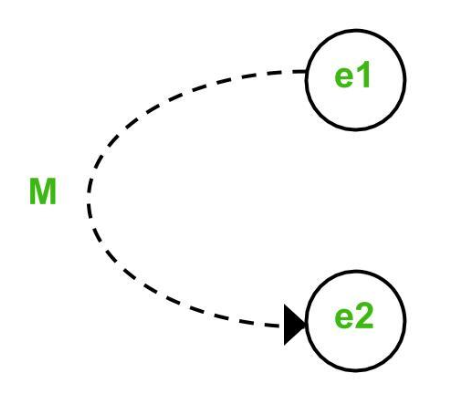
1. **Inclusive constraint**or **I-constraint:** This constraint exists between causes. It states that at least one of c1, c2 and c3 must always be 1, i.e., c1, c2 and c3 cannot be 0 simultaneously.

I

1. **Requires constraint**or **R-constraint:** This constraint exists between causes. It states that for c1 to be 1, c2 must be 1. It is impossible for c1 to be 1 and c2 to be 0.



1. **Mask constraint** or **M-constraint:** This constraint exists between effects. It states that if effect e1 is 1, the effect e2 is forced to be 0.



## Test case for the CAUSE-EFFECT GRAPH TECHNIQUE:-

## Test cases for the Tip calculator:-

## Causes are:-

## C1:- Tip percentage exceed 100

## C2:- Total Amount exceeds 10000

## C3:- Total Amount less than 50

## C4:- Tip percentage less than or equal 10

Effects are:-

e1:- Message should display “Maximum limit exceeds”

e2:- Message should display “Maximum limit exceeds pay through card or UPI”

e3:- Message should display “Amount is to low pay through cash”

e4:- Message should display that “Normal Tip”

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Decision Table for Cause effect Graph | | | | | | | |  |
|  | | Causes | | | | Effects | | | |
| Test ID | | C1 | C2 | C3 | C4 | E 1 | E2 | E3 | E4 |
| 1 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 4 | | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 6 | | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 7 | | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 8 | | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 9 | | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 10 | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 11 | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 12 | | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 13 | | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 14 | | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 15 | | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 16 | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |