|  |  |  |  |
| --- | --- | --- | --- |
| Test\_id | Description | Expected input | Expected output |
| LLR\_1 \_ Arithmetic Operations | It contains all the basic arithmetic operations. |  |  |
|  | Addition:  1)The user input must be validated. The sign of the user input must also be validated. | Integer, integer | integer |
|  | 2) The floating point input must provide a floating point results. | Float, Float | Float |
|  | 3) A combination of floating point input and integer input must provide a floating point output. | Float, integer  Or  Integer, float | Float |
|  | 4) If result exceeds by 14 digits then display unit must give the result as out of bound or out of range. | Input 1=10 digits  Input 2=6 digits  Or  Input 1=6 digits  Input 2=10 digits | Out of range  Or  Out of bound. |
|  | 5) If the first input is a negative number and second input is positive number or vice-versa.  If the negative input is greater than the positive input then the output must be negative. | Input1= -ve greater  Input2= +ve smaller  Or  Input1= +ve smaller  Input2= -ve greater | Negative  Negative |
|  | Subtraction:  1)The user input must be validated. The sign of the user input must be validated. | Integer or floating point input.  Alphanumeric input | Pass  Error |
|  | 2) If both the input is of integer type or floating type then the output must be integer or floating type. | Integer, integer  Or  Float, float | Integer  float |
|  | 3)If both the input sign is negative then the output must be the additive of both the values. | Input1= -ve  Input2= -ve | Output=  -(input1+input2) |
|  | 4)If the result exceeds 14 digits then the display unit must show out of bound or out of range | Input1=more than 14 digits  Input2= more than 14 digits | Output= result out of bound. |
|  | Multiplication:  1) The user input must be validated. The sign of the user input must be validated. | Integer or floating point input.  Alphanumeric input | Pass  Error |
|  | 2) If both the input is of integer type or floating type then the output must be integer or floating type. | Integer, integer  Or  Float, float | Integer  float |
|  | 3)If both the values are negative the output must have a positive sign.  If one input is positive and other one is negative then the resultant must have negative sign. | Input1=-ve  Input2=-ve  Input1=+ve  Input2=-ve | Output=+ve  Output=-ve |
|  | Division:  1) The user input must be validated. The sign of the user input must be validated. | Integer or floating point input.  Alphanumeric input | Pass  Error |
|  | 2) If both the input is of integer type or floating type then the output must be integer or floating type. | Integer, integer  Or  Float, float | Integer  float |
|  | 3) If both the values are negative the output must have a positive sign.  If one input is positive and other one is negative then the resultant must have negative sign. | Input1=-ve  Input2=-ve  Input1=+ve  Input2=-ve | Output=+ve  Output=-ve |
|  | 4) If the denominator is zero then the display unit must show error.  If the numerator is zero it must display infinite. | Input1 = digit  Input2 = zero  Input1=zero  Input2= digit | Error  infinite |
| LLR\_6\_Memory\_Storage | 1)It must display the last five results when the user hits the history button. | History | Last five results |
|  | 2) The history operation starts storing the results from first after switching on the calculator. | OFF  ON  History | No result  All the results are removed. |