Client-side input value for test case below

1. character – expects to receive a fail code.

2. one digit – expects to receive a fail code.

3. one digit with operator – expects to receive a fail code.

4. operator with a digit – expects to receive a fail code.

5. invalid operator (+) in between digits with space – expects to receive a fail code.

6. invalid operator (-) in between digits with space – expects to receive a fail code.

7. invalid operator (*) in between digits with space – expects to receive a fail code.

8. invalid operator (/) in between digits with space – expects to receive a fail code.

9. valid operator (+) in between digits with space – expects to receive a success code.

10. valid operator (-) in between digits with space – expects to receive a success code.

11. valid operator (*) in between digits with space – expects to receive a success code.

12. valid operator (/) in between digits with space – expects to receive a success code.

Test cases

1. character – expects to receive a fail code.

   Client side

   Input : test

   <<== At Client message to send out: 'test

   Status_Code received : 300

   Invalid input!!!

   Server side

   ==>> At Server receved message is: 'test'

   ==>> clientAddress is: 127.0.0.1/49832

   <<-- Server send back the result: '-1'

2. one digit – expects to receive a fail code.

   Client side

   <<== At Client message to send out: '2

   Status_Code received : 300

   Invalid input!!!

   Server side

   ==>> At Server receved message is: '2'

   ==>> clientAddress is: 127.0.0.1/49833

   <<-- Server send back the result: '-1'

3. one digit with operator – expects to receive a fail code.

   Client side

   <<== At Client message to send out: '5+

   Status_Code received : 300

   Invalid input!!!

   Server side

   ==>> At Server receved message is: '5+'

   ==>> clientAddress is: 127.0.0.1/49834

   <<-- Server send back the result: '-1'

4. operator with a digit – expects to receive a fail code.

Client side

<<== At Client message to send out: '* 4

Status_Code received : 300

Invalid input!!!

Server side

==>> At Server receved message is: '* 4'

==>> clientAddress is: 127.0.0.1/49835

<<-- Server send back the result: '-1'

5. invalid operator (%) in between digits with space – expects to receive a fail code.

Client side

<<== At Client message to send out: '4 % 2

Status_Code received : 300

Invalid input!!!

Server side

==>> At Server receved message is: '4 % 2'

==>> clientAddress is: 127.0.0.1/49836

<<-- Server send back the result: '-1'

6. valid operator (+) in between digits with/without space – expects to receive a success code.

Client side

<<== At Client message to send out: '4 +4

Status_Code received : 200

Result received! ==>> 4 +4: 8

Server side

==>> At Server receved message is: '4 +4'

==>> clientAddress is: 127.0.0.1/49837

<<== Server send back the result: '8'

7. valid operator (-) in between digits with/without space – expects to receive a success code.

Client side

<<== At Client message to send out: '4 -4

Status_Code received : 200

Result received! ==>> 4 -4: 0

Server side

==>> At Server receved message is: '4 -4'

==>> clientAddress is: 127.0.0.1/498328

<<== Server send back the result: '0'

8. valid operator (*) in between digits with/without space – expects to receive a success code.
   Client side
   <<== At Client message to send out: '4 *4
   Status_Code received : 200
   Result received! ==>> 4 *4: 16
   Server side
   ==>> At Server receved message is: '4 *4'
   ==>> clientAddress is: 127.0.0.1/49839
   <<== Server send back the result: '16'

9. valid operator (/) in between digits with/without space – expects to receive a success code.
   Client side
   <<== At Client message to send out: '4 /4
   Status_Code received : 200
   Result received! ==>> 4 /4: 1.0
   Server side
   ==>> At Server receved message is: '4 /4'
   ==>> clientAddress is: 127.0.0.1/49840
   <<== Server send back the result: '1.0'