|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version 3** | | | | | **Run by: Date:** | ***????? Tester's Name ????? ????? Date ?????*** |
| **Description** | **+ / − Purpose** | **Data Input** | **Expected result** | **Actual result if unexpected** | **Result** | **Comments** |
| converting() | + Normal Case: gives the same integer output when input is given as an input. | An integer number is given as an input. Input: 1234 | The same integer is expected as the output. Output: 1234 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when a float is given as the input. | An double number is given as an input. Input: 12.34 | The part after decimal point is truncated and only the number before decimal is expected as the output. Output: 12 |  | PASS |  |
| converting() | + Normal Case: zero as output when a float between 0 and 1 is given as the input. | An double number between zero and 1 is given as an input. Input: 0.314 | The part after decimal point is truncated and only zero is expected as the output. Output: 0 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float is given as the input. | A negative integer is given as an input. Input: -1234 | The same negative integer is expected as the output. Output: -1234 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float is given as the input. | A negative double is given as an input. Input: -12.34 | The part after decimal point is truncated and only the negative number before decimal is expected as the output. Output: -12 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when text/string is given as the input. | A text/string is given as an input. Input: hello | The output is zero because the data type is not numeric. Output: 0 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when a character is given as the input. | A character is given as an input. Input: @ | The output is zero because the data type is not numeric. Output: 0 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when an integer followed by string is given as the input. | An integer followed by string is given as the input. Input: 12 hello | The output is 12 and the string is not printed because it is not numeric. Output: 12 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when a float followed by string is given as the input. | A double followed by string is given as the input. Input: 12.34 hello | The output is 12 because the part after decimal point is truncated and string afterwards is not printed because it is not numeric. Output: 12 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float followed by string is given as the input. | A negative double followed by string is given as the input. Input: -12.34 hello | The output is -12 because the part after decimal point is truncated and string afterwards is not printed because it is not numeric. Output: -12 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when a string followed by integer is given as the input. | An string followed by integer is given as the input. Input: hello 1234 | The output is zero because the data type of first input is not numeric. Output: 0 |  | PASS |  |
| converting() | + Egde Case: gives expected output near/at the maximum size. | A double within input range being less than available memory. Input: 1234567891011121314.1111 | The output is 1234567891011121314 as expected where the double is converted to integer by truncating digits after decimal. Output: 1234567891011121314 |  | PASS |  |
| converting() | - Expecting a failed outcome by giving out of range positive input. | A number containing more than memory available is given as input. Input: 12345678910111213145.1111 | Expecting a garbage value as the output because of memory overflow. Output: 9223372036854775807 |  | FAIL | This test case failed because the range of the input was more than the memory available for the input. |
| converting() | + Egde Case: gives expected output near/at the minimun size. | A number within input range being more than available memory. Input: -1234567891011121314.1111 | The output is -1234567891011121314 as expected where the negative double is converted to integer by truncating digits after decimal Output: -1234567891011121314 |  | PASS |  |
| converting() | - Expecting a failed outcome by giving out of range negative input. | A negative number containing less than memory available is given as input. Input: -12345678910111213145.1111 | Expecting a garbage value as the output because of memory overflow. Output: -9223372036854775807 |  | FAIL | This test case failed because the range of the input was more than the memory available for the input. |
| converting() | + Normal Case: quits the code when 'q' is given as the input. | 'q' is given as the input. Input: q | The code is quit and an ending message is displayed. Output: \*\*\* End of Converting Strings to int Demo \*\*\* |  | PASS |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version 2** | | | | | **Run by: Date:** | ***Sampreet Klair 09-08-2023*** |
| **Description** | **+ / − Purpose** | **Data Input** | **Expected result** | **Actual result if unexpected** | **Result** | **Comments** |
| converting() | + Normal Case: gives the same integer output when input is given as an input. | An integer number is given as an input. Input: 1234 | A double is expected as the output with six digits after the decimal point. Output: 1234.000000 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when a float is given as the input. | A float number is given as an input. Input: 12.34 | A double is expected as the output with six digits after the decimal point. Output: 12.340000 |  | PASS |  |
| converting() | + Normal Case: zero as output when a float between 0 and 1 is given as the input. | A float number between zero and 1 is given as an input. Input: 0.314 | A double is expected as the output with six digits after the decimal point. Output: 0.314000 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float is given as the input. | A negative integer is given as an input. Input: -1234 | A negative double is expected as the output with six digits after the decimal point. Output: -1234.000000 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float is given as the input. | A negative float is given as an input. Input: -12.34 | A negative double is expected as the output with six digits after the decimal point. Output: -12.340000 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when text/string is given as the input. | A text/string is given as an input. Input: hello | The output is zero with six digits after the decimal point because the data type is not int. Output: 0.000000 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when a character is given as the input. | A character is given as an input. Input: @ | The output is zero with six digits after the decimal point because the data type is not int. Output: 0.000000 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when an integer followed by string is given as the input. | An integer followed by string is given as the input. Input: 12 hello | The output is 12 with six digits after the decimal point and the string is not printed because it is not int. Output: 12.000000 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when a float followed by string is given as the input. | A float followed by string is given as the input. Input: 12.34 hello | The output is a double with six digits after the decimal point and string afterwards is not printed because it is not int. Output: 12.340000 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float followed by string is given as the input. | A negative float followed by string is given as the input. Input: -12.34 hello | The output is a negative double with six digits after the decimal point and string afterwards is not printed because it is not int. Output: -12.340000 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when a string followed by integer is given as the input. | A string followed by integer is given as the input. Input: hello 1234 | The output is zero with six digits after decimal point because the data type of first input is not int. Output: 0.000000 |  | PASS |  |
| converting() | + Edge Case: gives expected output near/at the maximum size. | A number within input range being less than available memory. Input: 9999999999.1234 | A double is expected as the output with six digits after the decimal point. Output: 9999999999.123400 |  | PASS |  |
| converting() | - Expecting a failed outcome by giving out of range positive input. | A number containing more than 10 digits before decimal point. Input: 100000000000.1 | Expecting a garbage value after decimal point as the output because of memory overflow. Output: 100000000000.100006 |  | FAIL | This test case failed because the range of the input was more than the memory available for the input. |
| converting() | + Edge Case: gives expected output near/at the minimum size. | A number within input range being less than available memory. Input: -9999999999.1234 | A negative double is expected as the output with six digits after the decimal point. Output: -9999999999.123400 |  | PASS |  |
| converting() | - Expecting a failed outcome by giving out of range negative input. | A negative number containing more than 10 digits before decimal point. Input: -100000000000.1 | Expecting a garbage value as the output because of memory overflow. Output: -100000000000.100006 |  | FAIL | This test case failed because the range of the input was more than the memory available for the input. |
| converting() | + Normal Case: quits the code when 'q' is given as the input. | 'q' is given as the input. Input: q | The code is quit and an ending message is displayed. Output: \*\*\* End of Converting Strings to int Demo \*\*\* |  | PASS |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Program or module:** | *converting* | **Save As converting\_test\_cases** | | | | |
| **Version 1** | | | | | **Run by: Date:** | ***Sampreet Klair 05-08-2023*** |
| **Description** | **+ / − Purpose** | **Data Input** | **Expected result** | **Actual result if unexpected** | **Result** | **Comments** |
| converting() | + Normal Case: gives the same integer output when input is given as an input. | An integer number is given as an input. Input: 1234 | The same integer is expected as the output. Output: 1234 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when a float is given as the input. | A float number is given as an input. Input: 12.34 | The part after decimal point is truncated and only the number before decimal is expected as the output. Output: 12 |  | PASS |  |
| converting() | + Normal Case: zero as output when a float between 0 and 1 is given as the input. | A float number between zero and 1 is given as an input. Input: 0.314 | The part after decimal point is truncated and only zero is expected as the output. Output: 0 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float is given as the input. | A negative integer is given as an input. Input: -1234 | The same negative integer is expected as the output. Output: 1234 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float is given as the input. | A negative float is given as an input. Input: -12.34 | The part after decimal point is truncated and only the negative number before decimal is expected as the output. Output: -12 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when text/string is given as the input. | A text/string is given as an input. Input: hello | The output is zero because the data type is not int. Output: 0 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when a character is given as the input. | A character is given as an input. Input: @ | The output is zero because the data type is not int. Output: 0 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when an integer followed by string is given as the input. | An integer followed by string is given as the input. Input: 12 hello | The output is 12 and the string is not printed because it is not int. Output: 12 |  | PASS |  |
| converting() | + Normal Case: gives an integer as output when a float followed by string is given as the input. | A float followed by string is given as the input. Input: 12.34 hello | The output is 12 because the part after decimal point is truncated and string afterwards is not printed because it is not int. Output: 12 |  | PASS |  |
| converting() | + Normal Case: gives a negative integer as output when a negative float followed by string is given as the input. | A negative float followed by string is given as the input. Input: -12.34 hello | The output is -12 because the part after decimal point is truncated and string afterwards is not printed because it is not int. Output: -12 |  | PASS |  |
| converting() | + Normal Case: gives zero as output when a string followed by integer is given as the input. | A string followed by integer is given as the input. Input: hello 1234 | The output is zero because the data type of first input is not int. Output: 0 |  | PASS |  |
| converting() | + Edge Case: gives expected output near/at the maximum size. | A number within input range being less than available memory. Input: 999999999.1 | The output is 9999999999 as expected where the float is converted to integer by truncating digits after decimal Output: 999999999 |  | PASS |  |
| converting() | - Expecting a failed outcome by giving out of range positive input. | A number containing more than the range of input or more than memory available. Input: 10000000000.1 | Expecting a garbage value as the output because of memory overflow. Output: 1410065408 |  | FAIL | This test case failed because the range of the input was more than the memory available for the input. |
| converting() | + Edge Case: gives expected output near/at the minimum size. | A number within input range being less than available memory. Input: -999999999.1 | The output is -9999999999 as expected where the negative float is converted to integer by truncating digits after decimal Output: -999999999 |  | PASS |  |
| converting() | - Expecting a failed outcome by giving out of range negative input. | A negative number containing more than the range of input or more than memory available. Input: -10000000000.1 | Expecting a garbage value as the output because of memory overflow. Output: -1410065408 |  | FAIL | This test case failed because the range of the input was more than the memory available for the input. |
| converting() | + Normal Case: quits the code when 'q' is given as the input. | 'q' is given as the input. Input: q | The code is quit, and an ending message is displayed. Output: \*\*\* End of Converting Strings to int Demo \*\*\* |  | PASS |  |

**INPUT/OUPUT:**

**Version 3:**

\*\*\* Start of Converting Strings to int Demo \*\*\*

Type an int numeric string (q - to quit) :

q

\*\*\* End of Converting Strings to int Demo \*\*\*

\*\*\* Start of Converting Strings to double Demo \*\*\*

Type the double numeric string (q - to quit):

q

\*\*\* End of Converting Strings to double Demo \*\*\*

\*\*\* Start of Converting Strings to long Demo \*\*\*

Type the long numeric string (q - to quit):

1234

Converted number is 1234

Type the long numeric string (q - to quit):

12.34

Converted number is 12

Type the long numeric string (q - to quit):

0.314

Converted number is 0

Type the long numeric string (q - to quit):

-1234

Converted number is -1234

Type the long numeric string (q - to quit):

-12.34

Converted number is -12

Type the long numeric string (q - to quit):

hello

Converted number is 0

Type the long numeric string (q - to quit):

@

Converted number is 0

Type the long numeric string (q - to quit):

12 hello

Converted number is 12

Type the long numeric string (q - to quit):

12.34 hello

Converted number is 12

Type the long numeric string (q - to quit):

-12.34 hello

Converted number is -12

Type the long numeric string (q - to quit):

hello 1234

Converted number is 0

Type the long numeric string (q - to quit):

1234567891011121314.1111

Converted number is 1234567891011121314

Type the long numeric string (q - to quit):

12345678910111213145.1111

Converted number is 9223372036854775807

Type the long numeric string (q - to quit):

-1234567891011121314.1111

Converted number is -1234567891011121314

Type the long numeric string (q - to quit):

-12345678910111213145.1111

Converted number is -9223372036854775808

Type the long numeric string (q - to quit):

q

\*\*\* End of Converting Strings to long Demo \*\*\*

**Version 2:**

\*\*\* Start of Converting Strings to int Demo \*\*\*

Type an int numeric string (q - to quit) :

q

\*\*\* End of Converting Strings to int Demo \*\*\*

\*\*\* Start of Converting Strings to double Demo \*\*\*

Type the double numeric string (q - to quit):

1234

Converted number is 1234.000000

Type the double numeric string (q - to quit):

12.34

Converted number is 12.340000

Type the double numeric string (q - to quit):

0.314

Converted number is 0.314000

Type the double numeric string (q - to quit):

-1234

Converted number is -1234.000000

Type the double numeric string (q - to quit):

-12.34

Converted number is -12.340000

Type the double numeric string (q - to quit):

hello

Converted number is 0.000000

Type the double numeric string (q - to quit):

@

Converted number is 0.000000

Type the double numeric string (q - to quit):

12 hello

Converted number is 12.000000

Type the double numeric string (q - to quit):

12.34 hello

Converted number is 12.340000

Type the double numeric string (q - to quit):

-12.34 hello

Converted number is -12.340000

Type the double numeric string (q - to quit):

hello 1234

Converted number is 0.000000

Type the double numeric string (q - to quit):

9999999999.1234

Converted number is 9999999999.123400

Type the double numeric string (q - to quit):

100000000000.1

Converted number is 100000000000.100006

Type the double numeric string (q - to quit):

-9999999999.1234

Converted number is -9999999999.123400

Type the double numeric string (q - to quit):

100000000000.1

Converted number is 100000000000.100006

Type the double numeric string (q - to quit):

q

\*\*\* End of Converting Strings to double Demo \*\*\*

**Version 1:**

\*\*\* Start of Converting Strings to int Demo \*\*\*

Type an int numeric string (q - to quit) :

1234

Converted number is 1234

Type an int numeric string (q - to quit) :

12.34

Converted number is 12

Type an int numeric string (q - to quit) :

0.314

Converted number is 0

Type an int numeric string (q - to quit) :

-1234

Converted number is -1234

Type an int numeric string (q - to quit) :

-12.34

Converted number is -12

Type an int numeric string (q - to quit) :

hello

Converted number is 0

Type an int numeric string (q - to quit) :

@

Converted number is 0

Type an int numeric string (q - to quit) :

12 hello

Converted number is 12

Type an int numeric string (q - to quit) :

12.34 hello

Converted number is 12

Type an int numeric string (q - to quit) :

-12.34 hello

Converted number is -12

Type an int numeric string (q - to quit) :

hello 12.34

Converted number is 0

Type an int numeric string (q - to quit) :

999999999.1

Converted number is 999999999

Type an int numeric string (q - to quit) :

10000000000.1

Converted number is 1410065408

Type an int numeric string (q - to quit) :

-999999999.1

Converted number is -999999999

Type an int numeric string (q - to quit) :

-10000000000.1

Converted number is -1410065408

Type an int numeric string (q - to quit) :

q

\*\*\* End of Converting Strings to int Demo \*\*\*