|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Description | Input | Expected Result/Output |
| 1 | Outputs the product of the previous two fractions | (2/9)(4/3)\*# | Expression 1 is: (2/9)(4/3)\*  The value is: (8/27) |
| 2 | Outputs the sum of the two previous fractions | (1/1)(17/2)+# | Expression 2 is: (1/1)(17/2)+  The value is: (19/2) |
| 3 | Outputs the difference of the two previous fractions | (1/2)(1/4)-# | Expression 3 is: (1/2)(1/4)-  The value is: (1/4) |
| 4 | If nonexistent operator is entered | (2/3) B (1/3) / # | Expression 4 is: (2/3)B  Invalid Expression  Intermediate results: |
| 5 | If stack doesn’t have two fractions before the operator | (1/2) + (5/2) # | Expression 5 is: (1/2)+  Invalid Expression  Intermediate results: |
| 6 | If nothing is entered | # | Expression 7 is:  Invalid Expression  Intermediate results: |
| 7 | prints a normal termination message | End of File | Normal Termination of Program 3! |