Functions.java

1		`package calculator;`
2		
3		`import java.util.ArrayList;`
4		
5		`public class Functions {`
6		`    public static long factorial(double firstValue) {`
7	2 1. factorial : removed conditional - replaced comparison check with false → KILLED 2. factorial : changed conditional boundary → KILLED	`        if(firstValue < 0) {`
8		`            throw new IllegalArgumentException("Input must be a non-negative integer.");`
9		`        }`
10		`        long Answer;`
11	2 1. factorial : removed conditional - replaced equality check with false → SURVIVED 2. factorial : removed conditional - replaced equality check with false → KILLED	`        if(firstValue == 0 || firstValue == 1) {`
12		`            Answer = 1;`
13		`        } else {`
14		`            Answer = 1;`
15	2 1. factorial : removed conditional - replaced comparison check with false → KILLED 2. factorial : changed conditional boundary → KILLED	`            for (int i = 2; i <= firstValue; i++) {`
16	1 1. factorial : Replaced long multiplication with division → KILLED	`                Answer *= i;`
17		`            }`
18		`        }`
19	1 1. factorial : replaced long return with 0 for calculator/Functions::factorial → KILLED	`        return Answer;`
20		`    }`
21		
22		`    public static long sumFactorial(double firstValue, double secondValue) {`
23	2 1. sumFactorial : replaced long return with 0 for calculator/Functions::sumFactorial → KILLED 2. sumFactorial : Replaced long addition with subtraction → KILLED	`        return Functions.factorial(firstValue) + Functions.factorial(secondValue);`
24		`    }`
25		

26		`public static long divideFactorial(double firstValue, double secondValue) {`
27	2 1. divideFactorial : Replaced long division with multiplication → KILLED 2. divideFactorial : replaced long return with 0 for calculator/Functions::divideFactorial → KILLED	`    return Functions.factorial(firstValue) / Functions.factorial(secondValue);`
28		`}`
29		
30		`public static long summation(String equation, double firstValue, double secondValue, double thirdValue) {`
31		`    long start = (long) firstValue;`
32		`    long end = (long) secondValue;`
33		`    long constant = (long) thirdValue;`
34		`    long sum = 0;`
35		
36		`    switch(equation) {`
37		`        case "Cx":`
38	3 1. summation : Replaced long addition with subtraction → TIMED_OUT 2. summation : changed conditional boundary → KILLED 3. summation : removed conditional - replaced comparison check with false → KILLED	`            for (long n = start; n <= end; n++) {`
39	2 1. summation : Replaced long multiplication with division → KILLED 2. summation : Replaced long addition with subtraction → KILLED	`                sum += constant * n;`
40		`            }`
41		`            break;`
42		
43		`        case "x+C":`
44	3 1. summation : Replaced long addition with subtraction → TIMED_OUT 2. summation : removed conditional - replaced comparison check with false → KILLED 3. summation : changed conditional boundary → KILLED	`            for (long n = start; n <= end; n++) {`
45	2 1. summation : Replaced long addition with subtraction → KILLED 2. summation : Replaced long addition with subtraction → KILLED	`                sum += n + constant;`
46		`            }`
47		`            break;`
48		
49		`        case "x^C":`

50	3 1. summation : Replaced long addition with subtraction → TIMED_OUT 2. summation : removed conditional - replaced comparison check with false → KILLED 3. summation : changed conditional boundary → KILLED	`for (long n = start; n <= end; n++) {`
51	1 1. summation : Replaced double addition with subtraction → KILLED	`sum += Math.pow(n, constant);`
52		`}`
53		`break;`
54		
55		`default:`
56	2 1. summation : removed conditional - replaced equality check with false → SURVIVED 2. summation : removed conditional - replaced equality check with false → NO_COVERAGE	`if (constant == 0 || constant == 1) {`
57	3 1. summation : Replaced long addition with subtraction → TIMED_OUT 2. summation : removed conditional - replaced comparison check with false → KILLED 3. summation : changed conditional boundary → KILLED	`for (long n = start; n <= end; n++) {`
58	1 1. summation : Replaced long addition with subtraction → KILLED	`sum += n;`
59		`}`
60		`} else {`
61	3 1. summation : changed conditional boundary → NO_COVERAGE 2. summation : removed conditional - replaced comparison check with false → NO_COVERAGE 3. summation : Replaced long addition with subtraction → NO_COVERAGE	`for (long n = start; n <= end; n++) {`
62	1 1. summation : Replaced long addition with subtraction → NO_COVERAGE	`sum +=constant;`
63		`}`
64		`}`
65		`}`
66	1 1. summation : replaced long return with 0 for calculator/Functions::summation → KILLED	`return sum;`
67		`}`
68		
69		`public static long doubleSummation(int nValue, String equation, double firstValue, double secondValue, do`
70		`long start = (long) firstValue;`
71		`long end = (long) secondValue;`
72		

```
                                    long jstart = (long) thirdValue;
73                                  long jend = (long) fourthValue;
74                                  long sum = 0;
75
76  1 1. doubleSummation : removed conditional - replaced equality check with false → SURVIVED
                                    nValue = (nValue == 0) ? 1 : nValue;
77
78  3 1. doubleSummation : Replaced long addition with subtraction → TIMED_OUT
    2. doubleSummation : removed conditional - replaced comparison check with false → KILLED
    3. doubleSummation : changed conditional boundary → KILLED
                                    for(long n = start; n <= end; n++) {
79  3 1. doubleSummation : Replaced long addition with subtraction → TIMED_OUT
    2. doubleSummation : changed conditional boundary → KILLED
    3. doubleSummation : removed conditional - replaced comparison check with false → KILLED
                                        for(long j = jstart; j <= jend; j++) {
80                                          switch (equation) {
81                                              case "xy":
82  3 1. doubleSummation : Replaced long addition with subtraction → KILLED
    2. doubleSummation : Replaced long multiplication with division → KILLED
    3. doubleSummation : Replaced long multiplication with division → KILLED
                                                  sum += (nValue * n * j);
83                                                break;
84                                              case "x+y":
85  3 1. doubleSummation : Replaced long addition with subtraction → NO_COVERAGE
    2. doubleSummation : Replaced long addition with subtraction → NO_COVERAGE
    3. doubleSummation : Replaced long multiplication with division → NO_COVERAGE
                                                  sum += (nValue * n + j);
86                                                break;
87                                              case "x^y":
88  2 1. doubleSummation : Replaced double addition with subtraction → KILLED
    2. doubleSummation : Replaced double multiplication with division → KILLED
                                                  sum += (nValue * Math.pow(n, j));
89                                                break;
90                                              default:
91  2 1. doubleSummation : Replaced long addition with subtraction → NO_COVERAGE
```

91	2. doubleSummation : Replaced long multiplication with division → NO_COVERAGE	sum += n * j;
92		}
93		}
94		}
95	1 1. doubleSummation : replaced long return with 0 for calculator/Functions::doubleSummation → KILLED	return sum;
96		}
97		
98		public static long prodnot(String equation, double firstValue, double secondValue, double thirdValue) {
99		long start = (long) firstValue;
100		long end = (long) secondValue;
101		long constant = (long) thirdValue;
102		long sum = 1;
103		
104		switch(equation) {
105		case "Cx":
106	3 1. prodnot : Replaced long addition with subtraction → TIMED_OUT 2. prodnot : changed conditional boundary → KILLED 3. prodnot : removed conditional - replaced comparison check with false → KILLED	for (long n = start; n <= end; n++) {
107	2 1. prodnot : Replaced long multiplication with division → KILLED 2. prodnot : Replaced long multiplication with division → KILLED	sum *= constant * n;
108		}
109		break;
110		
111		case "x+C":
112	3 1. prodnot : Replaced long addition with subtraction → TIMED_OUT 2. prodnot : changed conditional boundary → KILLED 3. prodnot : removed conditional - replaced comparison check with false → KILLED	for (long n = start; n <= end; n++) {
113	2 1. prodnot : Replaced long multiplication with division → KILLED 2. prodnot : Replaced long addition with subtraction → KILLED	sum *= n + constant;
114		}

```
115                                 break;
116
117                             case "x^C":
    3 1. prodnot : removed conditional - replaced comparison check with false → NO_COVERAGE
118 2. prodnot : Replaced long addition with subtraction → NO_COVERAGE
    3. prodnot : changed conditional boundary → NO_COVERAGE
    1 1. prodnot : Replaced double multiplication with division → NO_COVERAGE
119                                 for (long n = start; n <= end; n++) {
120                                     sum *= Math.pow(n, constant);
121                                 }
122                                 break;
123                             default:
    2 1. prodnot : removed conditional - replaced equality check with false → SURVIVED
    2. prodnot : removed conditional - replaced equality check with false → KILLED
    3 1. prodnot : Replaced long addition with subtraction → TIMED_OUT
    2. prodnot : changed conditional
124 boundary → KILLED
    3. prodnot : removed conditional - replaced comparison check with false → KILLED
    1 1. prodnot : Replaced long multiplication with division → KILLED
125                                 if (constant == 0 || constant == 1) {
126                                     for (long n = start; n <= end; n++) {
127                                         sum *= n;
128                                     }
    3 1. prodnot : changed conditional boundary → NO_COVERAGE
    2. prodnot : Replaced long addition with subtraction → NO_COVERAGE
    3. prodnot : removed conditional - replaced comparison check with false → NO_COVERAGE
    1 1. prodnot : Replaced long multiplication with division → NO_COVERAGE
129                                 } else {
130                                     for (long n = start; n <= end; n++) {
131                                         sum *= constant;
132                                     }
133                                 }
134
    1 1. prodnot : replaced long return with 0 for calculator/Functions::prodnot → KILLED
135                             return sum;
136                         }
137
138                         public static long doubleProdNot(int nValue, String equation, double firstValue, double secondValue, doub
```

```
139                                 long start = (long) firstValue;
140                                 long end = (long) secondValue;
141                                 long jstart = (long) thirdValue;
142                                 long jend = (long) fourthValue;
143                                 long sum = 1;
144
145 1 1. doubleProdNot : removed conditional - replaced equality check with false → SURVIVED
                                    nValue = (nValue == 0) ? 1 : nValue;
146
147 3 1. doubleProdNot : Replaced long addition with subtraction → TIMED_OUT
    2. doubleProdNot : changed conditional boundary → KILLED
    3. doubleProdNot : removed conditional - replaced comparison check with false → KILLED
                                    for(long n = start; n <= end; n++) {
148 3 1. doubleProdNot : Replaced long addition with subtraction → TIMED_OUT
    2. doubleProdNot : removed conditional - replaced comparison check with false → KILLED
    3. doubleProdNot : changed conditional boundary → KILLED
                                        for(long j = jstart; j <= jend; j++) {
149                                         switch (equation) {
150                                             case "xy":
151 3 1. doubleProdNot : Replaced long multiplication with division → KILLED
    2. doubleProdNot : Replaced long multiplication with division → KILLED
    3. doubleProdNot : Replaced long multiplication with division → KILLED
                                                sum *= (nValue * n * j);
152                                             break;
153                                             case "x+y":
154 3 1. doubleProdNot : Replaced long multiplication with division → NO_COVERAGE
    2. doubleProdNot : Replaced long addition with subtraction → NO_COVERAGE
    3. doubleProdNot : Replaced long multiplication with division → NO_COVERAGE
                                                sum *= (nValue * n + j);
155                                             break;
156                                             case "x^y":
157 2 1. doubleProdNot : Replaced double multiplication with division → NO_COVERAGE
    2. doubleProdNot : Replaced double multiplication with division → NO_COVERAGE
                                                sum *= (nValue * Math.pow(n, j));
158                                             break;
```

```
159                                 default:
160 2 1. doubleProdNot : Replaced long multiplication with division → NO_COVERAGE
    2. doubleProdNot : Replaced long multiplication with division → NO_COVERAGE
                                    sum *= n * j;
161                                 }
162                             }
163                         }
164 1 1. doubleProdNot : replaced long return with 0 for calculator/Functions::doubleProdNot → KILLED
                        return sum;
165                 }
166
167                 public static double basicCalculation(String operator, double firstValue, double secondValue) {
168                     double Answer = 0;
169                     switch(operator) {
170                         case "//":
171 1 1. basicCalculation : Replaced double division with multiplication → KILLED
                                Answer = firstValue / secondValue;
172                             Answer = (long) Answer;
173                             break;
174                         case "%":
175 1 1. basicCalculation : Replaced double modulus with multiplication → KILLED
                                Answer = firstValue % secondValue;
176                             break;
177                         case "x^y":
178                             Answer = Math.pow(firstValue, secondValue);
179                             break;
180                         case "numroot":
181 1 1. basicCalculation : Replaced double division with multiplication → KILLED
                                Answer = Math.pow(secondValue, 1.0 / firstValue);
182                     }
183
184 1 1. basicCalculation : replaced double return with 0.0d for calculator/Functions::basicCalculation → KILLED
                        return Answer;
185                 }
186                 public static double calculateResult(ArrayList<Double> numbers,  ArrayList<String> operators) {
```

187 double result = numbers.get(0);

2 1. calculateResult : removed conditional - replaced comparison check with false → KILLED
188 2. calculateResult : changed conditional boundary → KILLED
for (int i = 1; i < numbers.size(); i++) {

189 String operator = operators.get(0);

190 switch(operator) {

191 case "+":

1 1. calculateResult : Replaced double addition with subtraction → KILLED
192 result += numbers.get(i);

193 break;

194 case "-":

1 1. calculateResult : Replaced double subtraction with addition → NO_COVERAGE
195 result -= numbers.get(i);

196 break;

197 case "*":

1 1. calculateResult : Replaced double multiplication with division → KILLED
198 result *= numbers.get(i);

199 break;

200 case "÷":

1 1. calculateResult : Replaced double division with multiplication → NO_COVERAGE
201 result /= numbers.get(i);

202 break;

203 }

204 operators.remove(0);

205 }

1 1. calculateResult : replaced double return with 0.0d for calculator/Functions::calculateResult → KILLED
206 return result;

207 }

208 public static String formatString(double Answer) {

209 String formattedAnswer;

2 1. formatString : removed conditional - replaced equality check with false → SURVIVED
210 2. formatString : Replaced double modulus with multiplication → SURVIVED
if (Answer % 1 == 0) {

211 formattedAnswer = String.valueOf((long) Answer);

212 } else {

```
213                 formattedAnswer = String.format("%.6f", Answer).replaceAll("0*$", "").replaceAll("\\.$", "");
214             }
215 1. formatString : replaced return value with "" for calculator/Functions::formatString → KILLED
            return formattedAnswer;
216         }
217     }
```

Mutations

7

1.**1**
Location : factorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_negative()] rem
check with false → KILLED

2.**2**
Location : factorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_zero()] changed

11

1.**1**
Location : factorial
Killed by : none removed conditional - replaced equality check with false → SURVIVED

2.**2**
Location : factorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_positive()] rem
with false → KILLED

15

1.**1**
Location : factorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_positive()] rem
with false → KILLED

2.**2**
Location : factorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_positive()] chan

16

1.**1**
Location : factorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_positive()] Rep
KILLED

19

1.**1**
Location : factorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_positive()] repl
calculator/Functions::factorial → KILLED

23

1.**1**
Location : sumFactorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:sumFactorial_basic()] re
calculator/Functions::sumFactorial → KILLED

2.**2**
Location : sumFactorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:sumFactorial_basic()] R
KILLED

1.**1**
Location : divideFactorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:divideFactorial_basic()]
→ KILLED

27

2.**2**
Location : divideFactorial
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:divideFactorial_basic()]
calculator/Functions::divideFactorial → KILLED

38

1.**1**
Location : summation
Killed by : none Replaced long addition with subtraction → TIMED_OUT

2.**2**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_Cx()] chang

3.**3**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_Cx()] remov
with false → KILLED

39

1.**1**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_Cx()] Repla
KILLED

2.**2**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_Cx()] Repla
KILLED

44

1.**1**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_plus_C()]
check with false → KILLED

2.**2**
Location : summation
Killed by : none Replaced long addition with subtraction → TIMED_OUT

3.**3**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_plus_C()]

45

1.**1**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_plus_C()]
KILLED

2.**2**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_plus_C()]
KILLED

50

1.**1**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_power_C(
comparison check with false → KILLED

2.**2**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_power_C(

3.**3**
Location : summation
Killed by : none Replaced long addition with subtraction → TIMED_OUT

51

1.**1**
Location : summation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_power_C(
→ KILLED

56	1.**1** **Location :** summation **Killed by :** none removed conditional - replaced equality check with false → SURVIVED 2.**2** **Location :** summation **Killed by :** none removed conditional - replaced equality check with false → NO_COVERAGE
57	1.**1** **Location :** summation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_default_sum comparison check with false → KILLED 2.**2** **Location :** summation **Killed by :** none Replaced long addition with subtraction → TIMED_OUT 3.**3** **Location :** summation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_default_sum KILLED
58	1.**1** **Location :** summation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_default_sum → KILLED
61	1.**1** **Location :** summation **Killed by :** none changed conditional boundary → NO_COVERAGE 2.**2** **Location :** summation **Killed by :** none removed conditional - replaced comparison check with false → NO_COVERAGE 3.**3** **Location :** summation **Killed by :** none Replaced long addition with subtraction → NO_COVERAGE
62	1.**1** **Location :** summation **Killed by :** none Replaced long addition with subtraction → NO_COVERAGE
66	1.**1** **Location :** summation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_default_sum calculator/Functions::summation → KILLED
76	1.**1** **Location :** doubleSummation **Killed by :** none removed conditional - replaced equality check with false → SURVIVED
78	1.**1** **Location :** doubleSummation **Killed by :** none Replaced long addition with subtraction → TIMED_OUT 2.**2** **Location :** doubleSummation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_po comparison check with false → KILLED 3.**3** **Location :** doubleSummation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_po KILLED
	1.**1** **Location :** doubleSummation

Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_po
KILLED

79

2.**2**
Location : doubleSummation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_po
comparison check with false → KILLED

3.**3**
Location : doubleSummation
Killed by : none Replaced long addition with subtraction → TIMED_OUT

1.**1**
Location : doubleSummation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_xy()]
KILLED

82

2.**2**
Location : doubleSummation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_xy()]
→ KILLED

3.**3**
Location : doubleSummation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_xy()]
→ KILLED

1.**1**
Location : doubleSummation
Killed by : none Replaced long addition with subtraction → NO_COVERAGE

85

2.**2**
Location : doubleSummation
Killed by : none Replaced long addition with subtraction → NO_COVERAGE

3.**3**
Location : doubleSummation
Killed by : none Replaced long multiplication with division → NO_COVERAGE

1.**1**
Location : doubleSummation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_po
subtraction → KILLED

88

2.**2**
Location : doubleSummation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_po
division → KILLED

1.**1**
Location : doubleSummation
Killed by : none Replaced long addition with subtraction → NO_COVERAGE

91

2.**2**
Location : doubleSummation
Killed by : none Replaced long multiplication with division → NO_COVERAGE

1.**1**
Location : doubleSummation
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_po
calculator/Functions::doubleSummation → KILLED

95

1.**1**
Location : prodnot
Killed by : none Replaced long addition with subtraction → TIMED_OUT

2.**2**
Location : prodnot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_Cx()] changed c

106

3.**3**

	Location : prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_Cx()] removed false → KILLED
107	1.**1** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_Cx()] Replaced KILLED 2.**2** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_Cx()] Replaced KILLED
112	1.**1** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_x_plus_C()] cha 2.**2** **Location :** prodnot **Killed by :** none Replaced long addition with subtraction → TIMED_OUT 3.**3** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_x_plus_C()] rer check with false → KILLED
113	1.**1** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_x_plus_C()] Re KILLED 2.**2** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_x_plus_C()] Re KILLED
118	1.**1** **Location :** prodnot **Killed by :** none removed conditional - replaced comparison check with false → NO_COVERAGE 2.**2** **Location :** prodnot **Killed by :** none Replaced long addition with subtraction → NO_COVERAGE 3.**3** **Location :** prodnot **Killed by :** none changed conditional boundary → NO_COVERAGE
119	1.**1** **Location :** prodnot **Killed by :** none Replaced double multiplication with division → NO_COVERAGE
124	1.**1** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_default_n()] ren with false → KILLED 2.**2** **Location :** prodnot **Killed by :** none removed conditional - replaced equality check with false → SURVIVED
125	1.**1** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_default_n()] cha 2.**2** **Location :** prodnot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_default_n()] ren check with false → KILLED

3.**3**
Location : prodnot
Killed by : none Replaced long addition with subtraction → TIMED_OUT

126

1.**1**
Location : prodnot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_default_n()] Re
KILLED

129

1.**1**
Location : prodnot
Killed by : none changed conditional boundary → NO_COVERAGE

2.**2**
Location : prodnot
Killed by : none Replaced long addition with subtraction → NO_COVERAGE

3.**3**
Location : prodnot
Killed by : none removed conditional - replaced comparison check with false → NO_COVERAGE

130

1.**1**
Location : prodnot
Killed by : none Replaced long multiplication with division → NO_COVERAGE

135

1.**1**
Location : prodnot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_x_plus_C()] rep
calculator/Functions::prodnot → KILLED

145

1.**1**
Location : doubleProdNot
Killed by : none removed conditional - replaced equality check with false → SURVIVED

147

1.**1**
Location : doubleProdNot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] ch

2.**2**
Location : doubleProdNot
Killed by : none Replaced long addition with subtraction → TIMED_OUT

3.**3**
Location : doubleProdNot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] rer
check with false → KILLED

148

1.**1**
Location : doubleProdNot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] rer
check with false → KILLED

2.**2**
Location : doubleProdNot
Killed by : none Replaced long addition with subtraction → TIMED_OUT

3.**3**
Location : doubleProdNot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] ch

1.**1**
Location : doubleProdNot
Killed by : calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] Re
KILLED

2.**2**
Location : doubleProdNot

151	**Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] Re KILLED 3.**3** **Location :** doubleProdNot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] Re KILLED
154	1.**1** **Location :** doubleProdNot **Killed by :** none Replaced long multiplication with division → NO_COVERAGE 2.**2** **Location :** doubleProdNot **Killed by :** none Replaced long addition with subtraction → NO_COVERAGE 3.**3** **Location :** doubleProdNot **Killed by :** none Replaced long multiplication with division → NO_COVERAGE
157	1.**1** **Location :** doubleProdNot **Killed by :** none Replaced double multiplication with division → NO_COVERAGE 2.**2** **Location :** doubleProdNot **Killed by :** none Replaced double multiplication with division → NO_COVERAGE
160	1.**1** **Location :** doubleProdNot **Killed by :** none Replaced long multiplication with division → NO_COVERAGE 2.**2** **Location :** doubleProdNot **Killed by :** none Replaced long multiplication with division → NO_COVERAGE
164	1.**1** **Location :** doubleProdNot **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] rep calculator/Functions::doubleProdNot → KILLED
171	1.**1** **Location :** basicCalculation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_floorDiv()] R → KILLED
175	1.**1** **Location :** basicCalculation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_mod()] Repla KILLED
181	1.**1** **Location :** basicCalculation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_numroot()] R → KILLED
184	1.**1** **Location :** basicCalculation **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_floorDiv()] re calculator/Functions::basicCalculation → KILLED
188	1.**1** **Location :** calculateResult **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:calcResult_add()] remov with false → KILLED 2.**2**

	Location : calculateResult **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:calcResult_add()] chang
192	1.**1** **Location :** calculateResult **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:calcResult_add()] Repla KILLED
195	1.**1** **Location :** calculateResult **Killed by :** none Replaced double subtraction with addition → NO_COVERAGE
198	1.**1** **Location :** calculateResult **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:calcResult_chain()] Repl KILLED
201	1.**1** **Location :** calculateResult **Killed by :** none Replaced double division with multiplication → NO_COVERAGE
206	1.**1** **Location :** calculateResult **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:calcResult_add()] replac calculator/Functions::calculateResult → KILLED
210	1.**1** **Location :** formatString **Killed by :** none removed conditional - replaced equality check with false → SURVIVED 2.**2** **Location :** formatString **Killed by :** none Replaced double modulus with multiplication → SURVIVED
215	1.**1** **Location :** formatString **Killed by :** calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:format_int()] replaced re calculator/Functions::formatString → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NULL_RETURNS
- PRIMITIVE_RETURNS
- REMOVE_CONDITIONALS_EQUAL_ELSE
- REMOVE_CONDITIONALS_ORDER_ELSE
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_x_power_y()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleSummation_xy()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_Cx()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_default_n()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:format_int()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:format_removeZeros()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:format_sixDecimals()] (1 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:doubleProdNot_xy()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:calcResult_add()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:calcResult_chain()] (0 ms)

- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_power_C()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_x_plus_C()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_default_sum_n()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:divideFactorial_basic()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:summation_Cx()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:prodnot_x_plus_C()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_floorDiv()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:sumFactorial_basic()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_positive()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_one()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_zero()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:factorial_negative()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_mod()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_numroot()] (29 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_unknown()] (0 ms)
- calculator.FunctionsUnitTest.[engine:junit-jupiter]/[class:calculator.FunctionsUnitTest]/[method:basicCalc_power()] (0 ms)

Report generated by PIT 1.15.3