MyAtoi.java

```
1     package StringAlgorithms;
2
3     public class MyAtoi {
4             public int myAtoi(String s) {
5                     s = s.trim();
6                     char[] char_1 = s.toCharArray();
7                     String number = "";
8                     boolean negative = false;
9                     boolean zero = false;
10                    boolean isDigit = false;
11
12                    for (char ch : char_1) {
13 1                          if (Character.isDigit(ch)) {
14 3                                  if (number.length() > 1 && !isDigit) {
15                                            number = "0";
16                                            break;
17                                    }
18                                    isDigit = true;
19 1                                  if (zero) {
20                                            number = "0";
21                                            break;
22                                    }
23                                    switch (ch) {
24                                    case '0' -> number += ch;
25                                    case '1' -> number += ch;
26                                    case '2' -> number += ch;
27                                    case '3' -> number += ch;
28                                    case '4' -> number += ch;
29                                    case '5' -> number += ch;
30                                    case '6' -> number += ch;
31                                    case '7' -> number += ch;
32                                    case '8' -> number += ch;
33                                    case '9' -> number += ch;
34                                    }
35 2                          } else if (ch == '-' && !isDigit) {
36                                    number += "0";
37                                    negative = true;
38 2                          } else if (ch == '+' && !isDigit) {
39                                    number += "0";
40 2                          } else if (ch == '.' && isDigit) {
41                                    break;
42 1                          } else if (ch == '.') {
```

```
43                                      zero = true;
44                              } else {
45 1                                    if (!isDigit) {
46                                              number = "0";
47                                      }
48                                      break;
49                              }
50                      }
51
52 1                    if (!isDigit) {
53                              return 0;
54                      }
55
56                      number = number.replaceFirst("^0+(?!$)", "");
57
58 3                    if (number.length() > 10 && negative) {
59 1                            return -2147483648;
60 2                    } else if (number.length() > 10) {
61 1                            return 2147483647;
62 2                    } else if (number.length() == 10 && negative) {
63                              double db1 = Double.parseDouble(number);
64 2                            if (db1 >= 2147483648d) {
65 1                                    return -2147483648;
66                              }
67 1                    } else if (number.length() == 10) {
68                              double db1 = Double.parseDouble(number);
69 2                            if (db1 > (2147483647)) {
70 1                                    return 2147483647;
71                              }
72                      }
73
74 1                    if(negative){
75 2                            return Integer.parseInt(number)*-1;
76                      }
77
78 1                    return Integer.parseInt(number);
79              }
80  }
```

Mutations

```
13  1. negated conditional → KILLED
    1. changed conditional boundary → KILLED
14  2. negated conditional → KILLED
    3. negated conditional → KILLED
19  1. negated conditional → KILLED
35  1. negated conditional → KILLED
```

	2. negated conditional → KILLED
38	1. negated conditional → KILLED 2. negated conditional → KILLED
40	1. negated conditional → KILLED 2. negated conditional → KILLED
42	1. negated conditional → SURVIVED
45	1. negated conditional → KILLED
52	1. negated conditional → KILLED
58	1. changed conditional boundary → KILLED 2. negated conditional → KILLED 3. negated conditional → KILLED
59	1. replaced int return with 0 for StringAlgorithms/MyAtoi::myAtoi → KILLED
60	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
61	1. replaced int return with 0 for StringAlgorithms/MyAtoi::myAtoi → KILLED
62	1. negated conditional → KILLED 2. negated conditional → KILLED
64	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
65	1. replaced int return with 0 for StringAlgorithms/MyAtoi::myAtoi → KILLED
67	1. negated conditional → KILLED
69	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
70	1. replaced int return with 0 for StringAlgorithms/MyAtoi::myAtoi → KILLED
74	1. negated conditional → KILLED
75	1. Replaced integer multiplication with division → SURVIVED 2. replaced int return with 0 for StringAlgorithms/MyAtoi::myAtoi → KILLED
78	1. replaced int return with 0 for StringAlgorithms/MyAtoi::myAtoi → KILLED

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- StringAlgorithms.AllStringTesting.[engine:junit-jupiter]/ [class:StringAlgorithms.AllStringTesting]/[method:testMyAtoi()] (11 ms)
- StringAlgorithms.AllStringTesting.[engine:junit-jupiter]/ [class:StringAlgorithms.AllStringTesting]/[method:letterCombinationsOfPhoneNumber()] (31 ms)

Report generated by PIT 1.6.8