

```
1 import components.simplewriter.SimpleWriter;
2
3
4 /**
5  * Compute integer power with interval halving and test it.
6  *
7  * @author Put your name here
8  *
9  */
10 public final class IntegerRoot {
11
12     /**
13      * Private constructor so this utility class cannot be instantiated.
14      */
15     private IntegerRoot() {
16     }
17
18     /**
19      * Returns {@code n} to the power {@code p}.
20      *
21      * @param n
22      *         the number to which we want to apply the power
23      * @param p
24      *         the power
25      * @return the number to the power
26      * @requires Integer.MIN_VALUE <= n ^ (p) <= Integer.MAX_VALUE and p >= 0
27      * @ensures power = n ^ (p)
28      */
29     private static int power(int n, int p) {
30         int result = 1, count = 0;
31         while (count < p) {
32             result *= n;
33             count++;
34         }
35         return result;
36     }
37
38     /**
39      * Returns the {@code r}-th root of {@code n}.
40      *
41      * @param n
42      *         the number to which we want to apply the root
43      * @param r
44      *         the root
45      * @return the root of the number
46      * @requires n >= 0 and r > 0
47      * @ensures root ^ (r) <= n < (root + 1) ^ (r)
48      */
49     private static int root(int n, int r) {
50         int low = 0, high = n + 1;
51         int guess;
52
53         while (high - low != 1) {
54             guess = (high + low) / 2;
55             if (n >= power(guess, r)) {
56                 low = guess;
57             } else {
58                 high = guess;
59             }
60         }
61         return low;
62     }
63 }
```

```
59     }
60 }
61     return low;
62 }
63
64 /**
65  * Main method.
66  *
67  * @param args
68  *     the command line arguments
69  */
70 public static void main(String[] args) {
71     SimpleWriter out = new SimpleWriter1L();
72
73     final int[] numbers = { 0, 0, 0, 1, 1, 1, 82, 82, 82, 82, 82, 3, 9, 27,
74                           81, 243 };
75     final int[] roots = { 1, 2, 5, 1, 2, 17, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5 };
76     final int[] results = { 0, 0, 0, 1, 1, 1, 82, 9, 4, 3, 2, 3, 3, 3, 3,
77                             3 };
78
79     for (int i = 0; i < numbers.length; i++) {
80         int x = root(numbers[i], roots[i]);
81         if (x == results[i]) {
82             out.println("Test passed: root(" + numbers[i] + ", " + roots[i]
83                         + ") = " + results[i]);
84         } else {
85             out.println("*** Test failed: root(" + numbers[i] + ", "
86                         + roots[i] + ") expected <" + results[i] + "> but was <"
87                         + x + ">");
88         }
89     }
90
91     out.close();
92 }
93
94 }
```