

```

1 import components.simplereader.SimpleReader;
2
3 /**
4  * Put a short phrase describing the program here.
5  *
6  * @author Put your name here
7  *
8  */
9 public final class ABCDGuesser2 {
10
11     /**
12      * No argument constructor--private to prevent instantiation.
13      */
14     private ABCDGuesser2() {
15
16     }
17
18     /**
19      * Main method.
20      *
21      * @param args
22      *         the command line arguments
23      */
24     public static void main(String[] args) {
25         SimpleReader in = new SimpleReader1L();
26         SimpleWriter out = new SimpleWriter1L();
27         double[] exp = { -5, -4, -3, -2, -1, -0.5, -0.333, -0.25, 0, 0.25,
28             0.333, 0.5, 1, 2, 3, 4, 5 };
29         double total = -1, closest = -999999999;
30         double bestW = -1, bestX = -1, bestY = -1, bestZ = -1;
31
32         // get u
33         double u = getPositiveDouble(in, out);
34
35         // get w x y z
36         double w = getPositiveDoubleNotOne(in, out);
37         double x = getPositiveDoubleNotOne(in, out);
38         double y = getPositiveDoubleNotOne(in, out);
39         double z = getPositiveDoubleNotOne(in, out);
40
41         // big boy loop
42         for (int i = 0; i <= 16; i++) {
43             for (int j = 0; j <= 16; j++) {
44                 for (int k = 0; k <= 16; k++) {
45                     for (int l = 0; l <= 16; l++) {
46                         // does calculations
47                         total = Math.pow(w, exp[i]);
48                         total += Math.pow(x, exp[j]);
49                         total += Math.pow(y, exp[k]);
50                         total += Math.pow(z, exp[l]);
51
52                         // if total is the closest to u so far
53                         if (Math.abs(u - total) < Math.abs(u - closest)) {
54                             closest = total;
55                             bestW = exp[i];
56                             bestX = exp[j];
57                             bestY = exp[k];
58                             bestZ = exp[l];
59                         }
60                     }
61                 }
62             }
63         }
64     }
65 }

```

```

64         }
65     }
66 }
67
68 // do some math
69 double percentError = Math.abs((closest - u) / u * 100);
70
71 // print results
72 System.out.println("u = " + u);
73 System.out.println("(" + w + "^" + bestW + ")" + " + " + "(" + x + "^"
74     + bestX + ")" + " + " + "(" + y + "^" + bestY + ")" + " + "
75     + "(" + z + "^" + bestZ + ")" + " = "
76     + String.format("%.2f", closest));
77 System.out.println(
78     "Percent Error: " + String.format("%.2f", percentError) + "%");
79
80 // close stuff
81 in.close();
82 out.close();
83 }
84
85 /**
86  * Repeatedly asks the user for a positive real number until the user enters
87  * one. Returns the positive real number.
88  *
89  * @param in
90  *     the input stream
91  * @param out
92  *     the output stream
93  * @return a positive real number entered by the user
94  */
95 private static double getPositiveDouble(SimpleReader in, SimpleWriter out) {
96     System.out.print("Enter a positive number: ");
97     String input = in.nextLine();
98     boolean looping = true;
99     double num = -1;
100
101     while (looping) {
102         if (FormatChecker.canParseInt(input)) {
103             num = Integer.parseInt(input);
104             if (num > 0) {
105                 looping = false;
106             }
107         } else {
108             System.out.print("Enter a positive number: ");
109             input = in.nextLine();
110         }
111     }
112
113     return num;
114 }
115
116 /**
117  * Repeatedly asks the user for a positive real number not equal to 1.0
118  * until the user enters one. Returns the positive real number.
119  *
120  * @param in
121  *     the input stream
122  * @param out

```

```
123      *           the output stream
124      * @return a positive real number not equal to 1.0 entered by the user
125      */
126      private static double getPositiveDoubleNotOne (SimpleReader in,
127      SimpleWriter out) {
128          boolean looping = true;
129          double num = -1;
130
131          while (looping) {
132              num = getPositiveDouble(in, out);
133              if (num != 1.0) {
134                  looping = false;
135              }
136          }
137
138          return num;
139      }
140
141
142
```