Test (28-8-24)

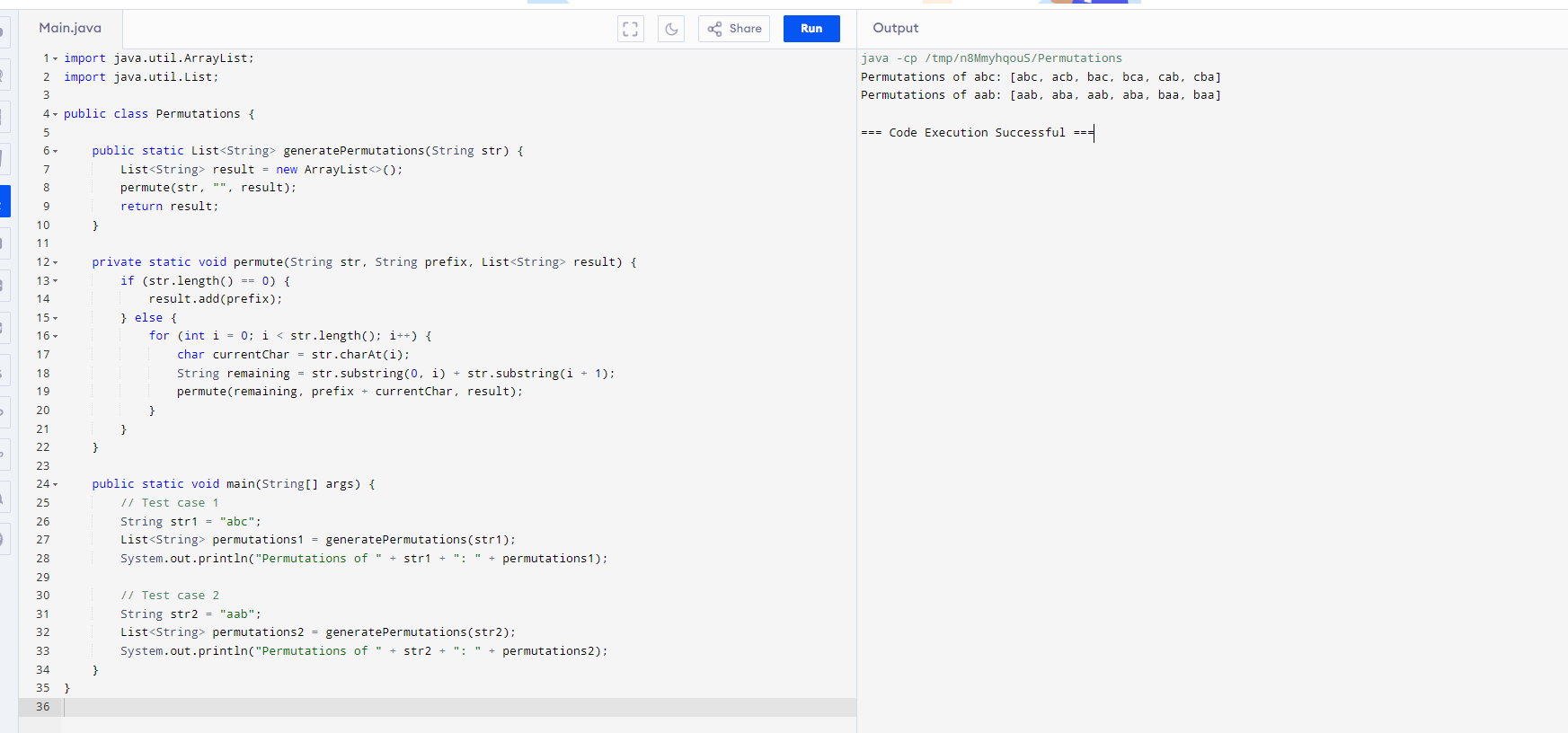
1. Write a recursive method to generate all permutations of a given string.

Test Cases:

&quot;abc&quot; → [abc, bac, bca, cab, cba]

&quot;aab&quot; → [aab, aba, baa]

Program:&output:



2. Write a Java method that extracts all valid URLs from a given string. A valid URL must start with http or https, followed by ://, and should contain a valid domain name.

Test Cases:

String text = &quot;Visit https://www.example.com and http://www.test.com for

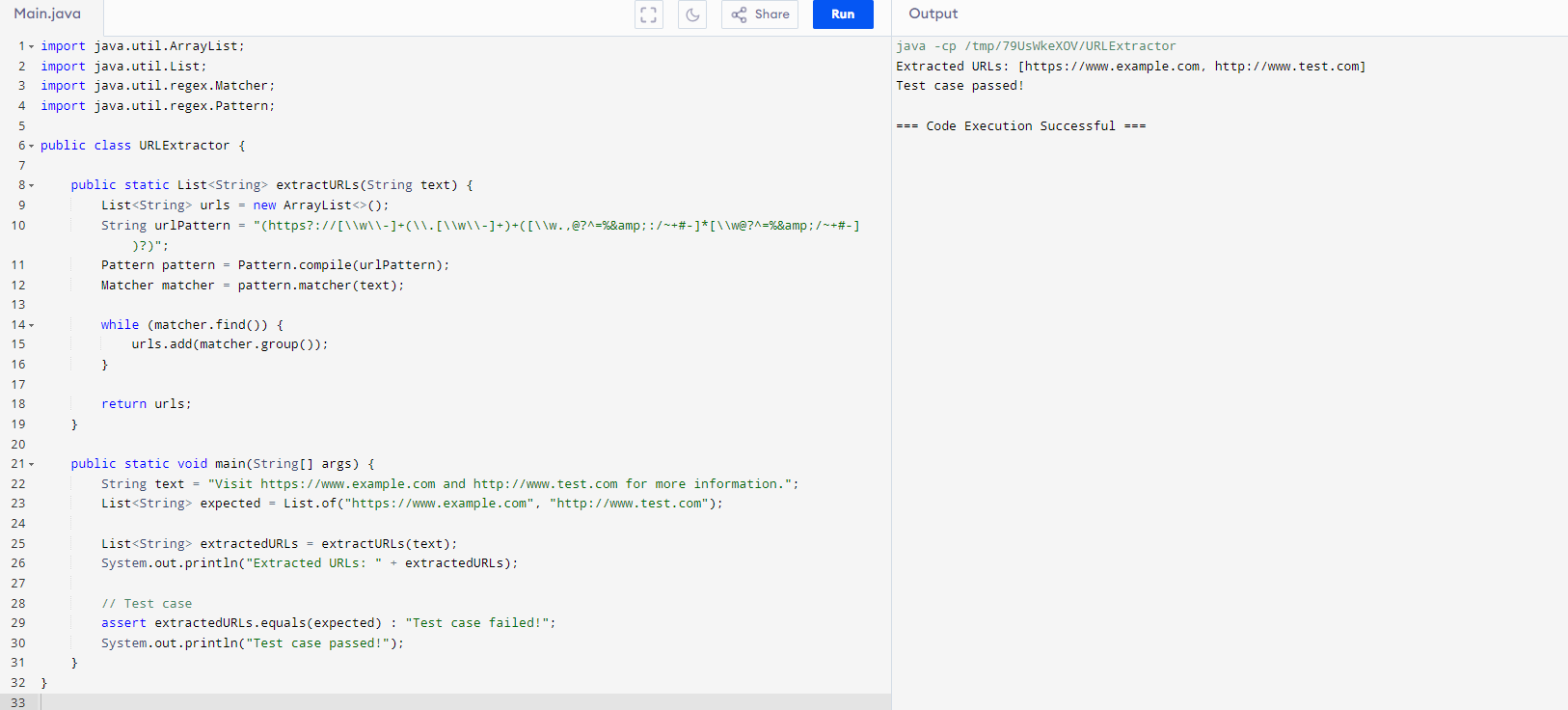
more information.&quot;;

List&lt;String&gt; expected = Arrays.asList(&quot;https://www.example.com&quot;,

&quot;http://www.test.com&quot;);

assert extractURLs(text).equals(expected); // true

program:&output;



3. Write a Java method to validate if a given string represents a valid time inthe 24-hour format (HH).

Test Cases:

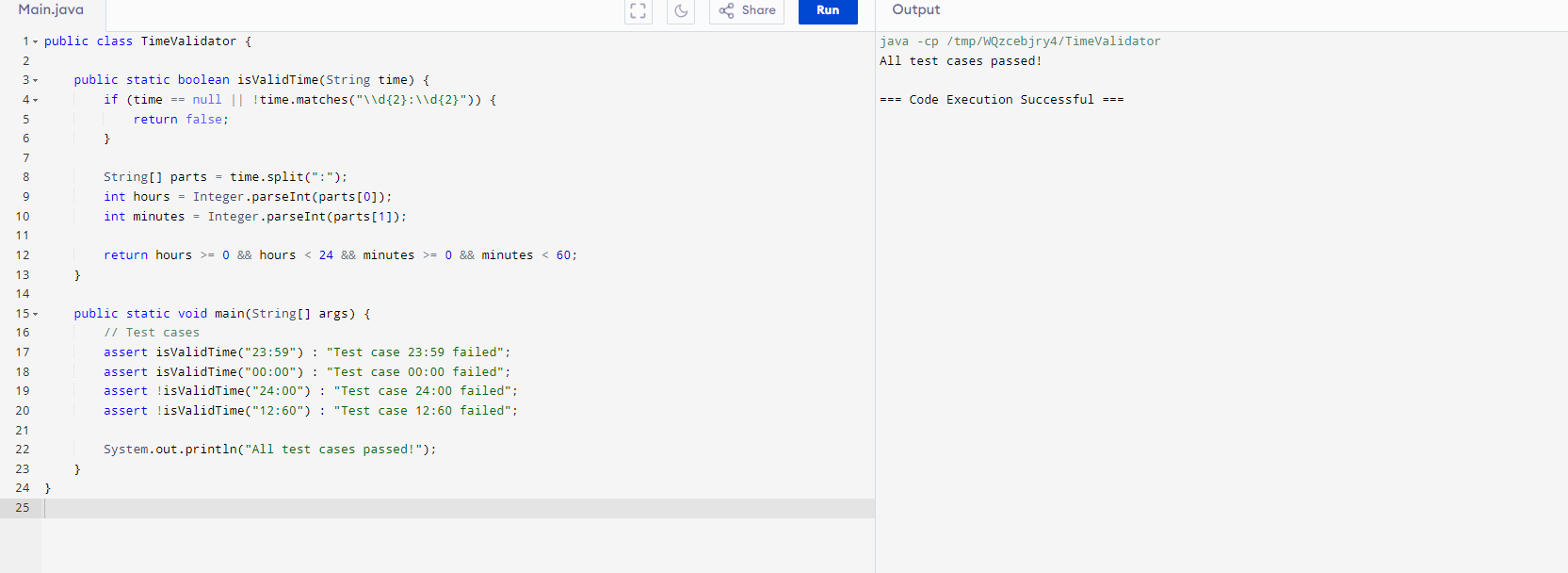
assert isValidTime(&quot;23:59&quot;); // true

assert isValidTime(&quot;00:00&quot;); // true

assert !isValidTime(&quot;24:00&quot;); // false

assert !isValidTime(&quot;12:60&quot;); // false

program;&output:



4. Write a recursive Java method to generate all subsets of a given set of

integers.

Test Cases:

List&lt;Integer&gt; set = Arrays.asList(1, 2, 3);

List&lt;List&lt;Integer&gt;&gt; expected = Arrays.asList(

Arrays.asList(),

Arrays.asList(1),

Arrays.asList(2),

Arrays.asList(3),

Arrays.asList(1, 2),

Arrays.asList(1, 3),

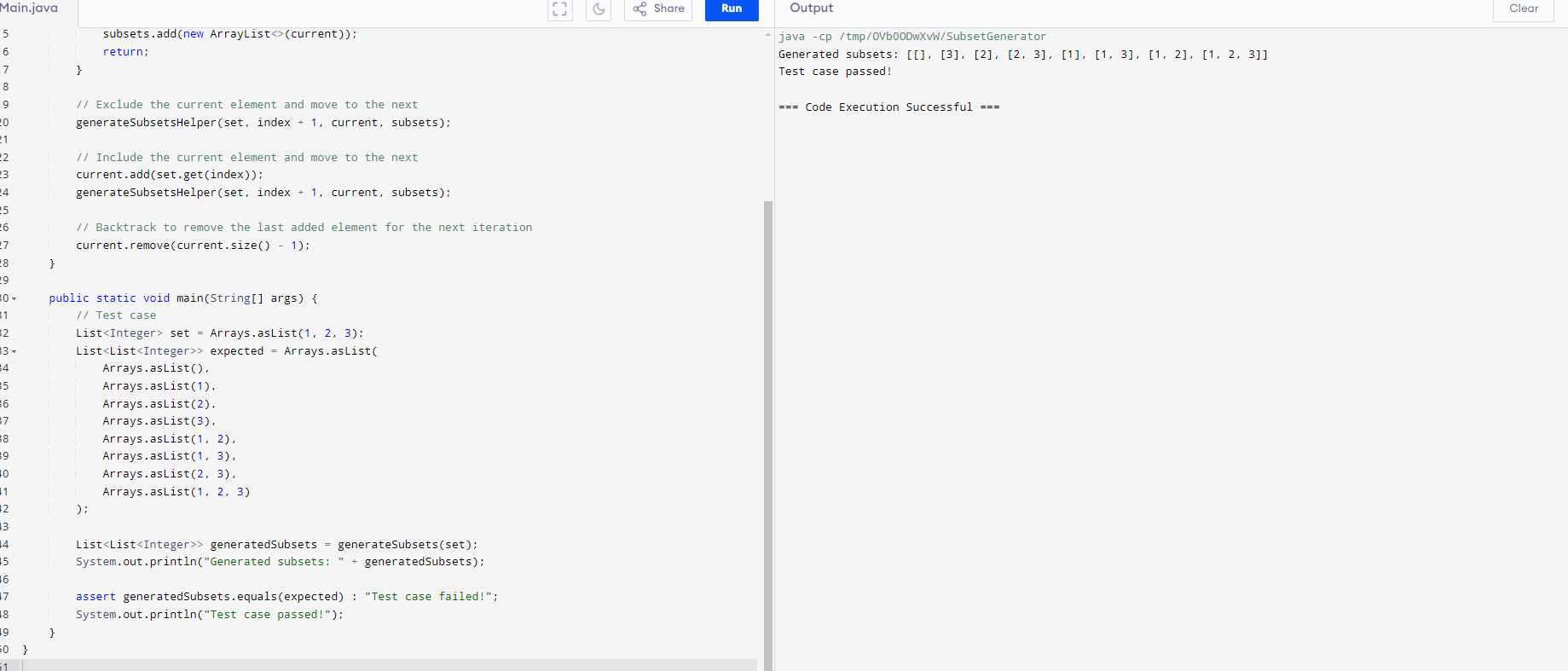
Arrays.asList(2, 3),

Arrays.asList(1, 2, 3)

);

assert generateSubsets(set).equals(expected); // true

program:&output:



5. Write a recursive Java method to determine if a string can be segmented

into a space-separated sequence of one or more dictionary words.

Hard

Test Cases:

Set&lt;String&gt; wordDict = new HashSet&lt;&gt;(Arrays.asList(&quot;apple&quot;, &quot;pen&quot;,

&quot;applepen&quot;, &quot;pine&quot;, &quot;pineapple&quot;));

assert wordBreak(&quot;pineapplepenapple&quot;, wordDict) == true; // true

assert wordBreak(&quot;catsandog&quot;, wordDict) == false; // true

program:&output:

