**TESTING PART 1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 1 A | letterInOriginal | letterInOriginal 'A' [('A','B'),('C','D')] | True | PASSED: |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 1 B | letterInOriginal | letterInOriginal 'X' [('A','B'),('C','D')] | False | PASSED: |

**TESTING PART 2A:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 2 A | findLetterPos | findLetterPos ('A','B') [('A','B'),('C','D')] | 0 | PASSED: |

**TESTING PART 2B:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 2 B -1 | forwardFind | forwardFind ('W','R', []) testCrib | [0,5,21,12,7…] | FAILED: |

This test had everything correct except the first position was never added, which is must be in the final result however, this problem should be an easy fix by simply adding the original position into the inputted list.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 2 B- 2 | forwardFind | forwardFind ('W','R', [0]) testCrib | [0,5,21,12,7…] | PASSED: |

**TESTING PART 3A:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 3 A -1 | forwardFind | forwardFind ('W','R', [5,21,21,12,7 | (“Y”,”V”,[19 ..) | PASSED |

**TESTING PART 3B:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 3B | getLongest | getLongest [[0,3,2], [3,4,23,525,-1]] | [3,4,23,525,-1]] | PASSED: |

**TESTING PART 3C:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 3C | usingForwardFind | usingForwardFind MAINCRIB | 3,6,5… | PASSED: |

**TESTING PART 4A:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | CODE TESTED: | INPUT VALUES: | EXPECTED: | RESULT: |
| 4A | giveRandom  Element | giveRandom  Element 34 [(A,B), (C,D), (E,F)] 0 | (A,B) | PASSED :  (A,B) |

Although possibly logically flawed code, this works really well in practice.