Our Solution(s) Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class Program {
  // O(nd) time | O(n) space
  public static int numberOfWaysToMakeChange(int n, int[] denoms
    int[] ways = new int[n + 1];
    ways[0] = 1;
    for (int denom : denoms) {
      for (int amount = 1; amount < n + 1; amount++) {
        if (denom <= amount) {
          ways[amount] += ways[amount - denom];
        }
      }
    }
    return ways[n];
  }
}

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
class Program {
  public static int numberOfWaysToMakeChange(int n, int[] denoms
    // Write your code here.
    return -1;
  }
}

```

Our Tests

Custom Output Submit Code

```
@Test
public void Test0(){
    int[] input = {3, 5, 8, 7};
    Utils.assertTrue(Program.numberOfWaysToMakeChange(8, input));
}

@Test
public void Test1(){
    int[] input = {2};
    Utils.assertTrue(Program.numberOfWaysToMakeChange(5, input));
}

@Test
public void Test2(){
    int[] input = {3, 4};
    Utils.assertTrue(Program.numberOfWaysToMakeChange(7, input));
}

@Test
public void Test3(){
    int[] input = {3, 5};
```

Run or submit code when you're ready.