Our Solution(s)   Run Code

Solution 1

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

#include <vector>
using namespace std;

// O(nd) time | O(n) space
int numberOfWaysToMakeChange(int n, vector<int> denoms) {
  vector<int> ways(n + 1, 0);
  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

```cpp
#include <vector>
using namespace std;

int numberOfWaysToMakeChange(int n, vector<int> denoms) {
  // Write your code here.
  return -1;
}
```

Our Tests

Custom Output   Submit Code

```
vector<int> vector1{0};
vector<int> vector2{1, 0};
vector<int> vector3{1, 7};
vector<int> vector4{1, 7, 10, 20};
vector<int> vector5{1, 7, 10, 20};
vector<int> vector6{1, 7, 10, 20};
vector<int> vector7{1, 7, 10, 20};
vector<int> vector8{1, 2, 7};
vector<int> vector9{1, 2, 4, 7};

class Program1 : public TestSuite {
public:
    void Run() {

        TestCase("Test Case 1");
                [&] { assertIsImNotEmptyIsInRange(0, vector1) }

        TestCase("Test Case 2");
                [&] { assertIsImNotEmptyIsInRange(0, vector2) }

        TestCase("Test Case 3");
```

Run or submit code when you're ready.