

Our Solution(s)Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.Arrays;
4
5 class Program {
6     // O(nd) time | O(n) space
7     public static int minNumberOfCoinsForChange(int n, int[] denoms) {
8         int[] numOfCoins = new int[n + 1];
9         Arrays.fill(numOfCoins, Integer.MAX_VALUE);
10        numOfCoins[0] = 0;
11        int toCompare = 0;
12        for (int denom : denoms) {
13            for (int amount = 0; amount < numOfCoins.length; amount++) {
14                if (denom <= amount) {
15                    if (numOfCoins[amount - denom] == Integer.MAX_VALUE) {
16                        toCompare = numOfCoins[amount - denom];
17                    } else {
18                        toCompare = numOfCoins[amount - denom] + 1;
19                    }
20                    numOfCoins[amount] = Math.min(numOfCoins[amount], toCompare);
21                }
22            }
23        }
24        return numOfCoins[n] != Integer.MAX_VALUE ? numOfCoins[n] : -1;
25    }
26 }
27
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 class Program {
2     public static int minNumberOfCoinsForChange(int n, int[] denoms) {
3         // Write your code here.
4         return -1;
5     }
6 }
7
```

```

38 #Print
39 #Print testCases() : 1
40 test() input = (5, 5, 10)
41 #Print testProgram testCases() testCases(5, input)
42 1
43
44 #Print
45 #Print testCases() : 1
46 test() input = (5, 5, 10)
47 #Print testProgram testCases() testCases(5, input)
48 1
49
50 #Print
51 #Print testCases() : 1
52 test() input = (5, 5, 10)
53 #Print testProgram testCases() testCases(5, input)
54 1
55
56 #Print
57 #Print testCases() : 1
58 test() input = (5, 5, 10)

```

Run or submit code when you're ready.