Solution 1

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

class Program {
    // O(nd) time | O(n) space
    func minimumNumberOfCoinsForChange(target: Int, denominations: [Int]) -> Int {
        var numberOfCoins = Array(repeating: Int(Int16.max), count: target + 1)
        numberOfCoins[0] = 0

        for denomination in denominations {
            for amount in 0 ..< numberOfCoins.count {
                if denomination <= amount {
                    numberOfCoins[amount] = min(numberOfCoins[amount], 1 + numberOfCoins[am
                }
            }
        }

        return numberOfCoins[target] != Int(Int16.max) ? numberOfCoins[target] : -1
    }
}
```

Solution 1   Solution 2   Solution 3

```swift
class Program {
    func minimumNumberOfCoinsForChange(target: Int, denominations: [Int]) -> Int {
        // Write your code here.
        return -1
    }
}
```

Custom Output   Raw Output   Submit Code

Run or submit code when you're ready.

Solution 1

```swift
class Program {
    func minimumNumberOfCoinsForChange(target: Int, denominations: [Int]) -> Int {
        var numberOfCoins = Array(repeating: Int(Int16.max), count: target + 1)
        numberOfCoins[0] = 0

        for denomination in denominations {
            for amount in 0 ..< numberOfCoins.count {
                if denomination <= amount {
                    numberOfCoins[amount] = min(numberOfCoins[amount], 1 + numberOfCoins[am
                }
            }
        }

        return numberOfCoins[target] != Int(Int16.max) ? numberOfCoins[target] : -1
```

Solution 1   Solution 2   Solution 3

```swift
class Program {
    func minimumNumberOfCoinsForChange(target: Int, denominations: [Int]) -> Int {
        // Write your code here.
        return -1
    }
}
```