

Our Solution(s)Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 import "math"
6
7 func MinNumberOfCoinsForChange(n int, denoms []int) int {
8     numOfCoins := make([]int, n+1)
9     for i := range numOfCoins {
10         numOfCoins[i] = math.MaxInt32
11     }
12     numOfCoins[0] = 0
13     for _, denom := range denoms {
14         for amount := range numOfCoins {
15             if denom <= amount {
16                 numOfCoins[amount] = min(numOfCoins[amount], numOfCoins[amount-1]+1)
17             }
18         }
19     }
20     if numOfCoins[n] != math.MaxInt32 {
21         return numOfCoins[n]
22     }
23     return -1
24 }
25
26 func min(arg1 int, rest ...int) int {
27     curr := arg1
28     for _, num := range rest {
29         if num < curr {
30             curr = num
31         }
32     }
33     return curr
34 }
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 package main
2
3 func MinNumberOfCoinsForChange(n int, denoms []int) int {
4     // Write your code here.
5     return -1
6 }
7
```

Custom OutputSubmit Code

```
1 import sys
2
3
4 def is_prime(n):
5     if n < 2:
6         return False
7     for i in range(2, int(n**0.5) + 1):
8         if n % i == 0:
9             return False
10    return True
11
12 def main():
13     n = int(sys.argv[1])
14     if is_prime(n):
15         print("Prime")
16     else:
17         print("Not Prime")
18
19 if __name__ == "__main__":
20     main()
```

Run or submit code when you're ready.