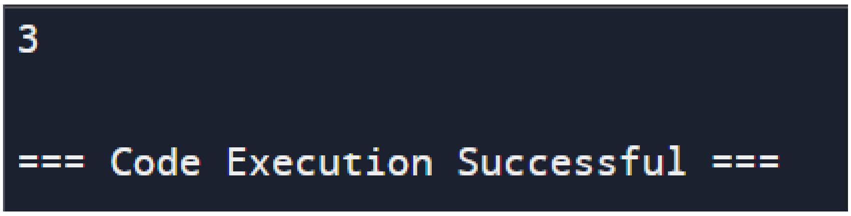
1. Coin Change Problem

PROGRAM:-

```
def coinChange(coins, amount):
    # Create a list to store the minimum number of coins for each amount up to the target amount.
    # Initialize the list with a value greater than any possible number of coins.
    dp = [float('inf')] * (amount + 1)

    # Base case: No coins are needed to make amount 0.
    dp[0] = 0

    # Loop through each coin and update the dp list accordingly.
    for coin in coins:
        for x in range(coin, amount + 1):
            dp[x] = min(dp[x], dp[x - coin] + 1)

    # If dp[amount] is still float('inf'), it means it's not possible to make that amount with the given coins.
    return dp[amount] if dp[amount] != float('inf') else -1

# Example usage:
coins = [1, 2, 5]
amount = 11
print(coinChange(coins, amount))  # Output: 3 (11 = 5 + 5 + 1)
```

OUTPUT:-



```
3

=== Code Execution Successful ===
```

TIME COMPLEXITY:-   O(n*m)