**Introduction:**

In this lab, we designed a solution for Change-Making problem by using Greedy algorithm and Dynamic Programming. Given a large number of 1, 5, 10 and 25 coins denominations, goal was to choose fewest coins totaling N, where N is the amount of money to be changed.

The change-making problem addresses the following question: how can a given amount of money be made with the least number of coins of given denominations? In simple words, it is the problem of representing a given value with the fewest coins possible. It is a knapsack type problem, and has applications wider than just currency. An application of change-making problem can be found in computing the ways one can make a nine dart finish in a game of darts.

**Analysis:**

The following table shows 10 different values for which the Dynamic Programming approach is more optimal than the Greedy Algorithm approach.

|  |  |  |
| --- | --- | --- |
| Values | Greedy Algorithm | Dynamic Programming |
| 25 | 1 | 1 |
| 33 | 5 | 5 |
| 50 | 2 | 2 |
| 67 | 6 | 6 |
| 99 | 6 | 6 |
| 125 | 5 | 5 |
| 156 | 8 | 8 |
| 177 | 9 | 9 |
| 233 | 13 | 13 |
| 299 | 17 | 17 |

Since there is a coin of denomination 5 available as well, there are no values for which the Dynamic Programming approach is better than the Greedy Algorithm approach. If the value of 5 was not available, Dynamic Programming approach would have been better than the Greedy Algorithm approach for numerous values.

**How to Run:**

Run the program. Choose the approach for coin change with the help of the given menu and entered the value which needs to be changed. The results will be displayed.