Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: DAA (01CT0512)	**AIM: Making Coin Change**	
Experiment No: 22	**Date: 3/10/2023**	**Enrolment No: 92100133020**

Making Coin Change:

Given a set of coins and a total amount, the problem is to find the minimum number of coins needed to make up that amount.

Algorithm:

1. Create an array **dp[amount + 1]** initialized with INT_MAX except **dp[0]** set to 0.
2. For each coin, update **dp[i]** for all **i** from coin value to amount, taking the minimum of **dp[i]** and **dp[i - coin] + 1**.

Code:

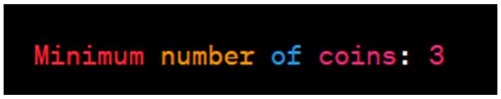
```
#include <iostream>
#include <climits>
using namespace std;

int minCoins(int coins[], int n, int amount) {
  int dp[amount + 1];
  fill(dp, dp + amount + 1, INT_MAX);
  dp[0] = 0;

  for (int i = 1; i <= amount; i++) {
    for (int j = 0; j < n; j++) {
      if (coins[j] <= i) {
        dp[i] = min(dp[i], dp[i - coins[j]] + 1);
      }
    }
  }
  return dp[amount];
}

int main() {
  int coins[] = {1, 2, 5};
  int n = sizeof(coins) / sizeof(coins[0]);
  int amount = 11;
  cout << "Minimum number of coins: " << minCoins(coins, n, amount);
  return 0;
}
```

Output:



```
Minimum number of coins: 3
```

Space complexity: ___________

Justification:____
____

Time complexity:

Best case time complexity: ___________

Justification:____
____

Worst case time complexity: ___________

Justification:____