East West University
Department of Computer Science and Engineering

Course: CSE 246(Algorithms)
Section - 02

Submitted To :	**Submitted By :**
Dr. Md. Tauhid Bin Iqbal Assistant Professor Department of CSE East West University	**Name** : Hafsa Ferdousi **Student ID** : 2023-3-60-321 **Department** : CSE **Date** : 09 / 12 / 2025

1.Minimum Coin change

```
#include <bits/stdc++.h>
using namespace std;

int coinChange(vector<int> &coins, int amount) {
    int n = coins.size();
    vector<vector<int>> dp(n + 1, vector<int>(amount + 1, INT_MAX - 1));

    for (int i = 0;i<= n; i++) {
        dp[i][0] = 0;
    }

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= amount; j++) {

            if (j >= coins[i - 1]) {
                dp[i][j] = min(dp[i - 1][j],
                               dp[i][j - coins[i - 1]] + 1);
            } else {
                dp[i][j] = dp[i - 1][j];
            }
        }
    }

    if (dp[n][amount] >= INT_MAX - 1)
        return -1;

    return dp[n][amount];
}

int main() {
    int n, amount;
    cout << "enter number of coins: ";
    cin >> n;

    vector<int> coins(n);
    cout << "Enter coin values: ";
    for (int i = 0; i < n; i++)
        cin >> coins[i];

    cout << "Enter amount: ";
    cin >> amount;
    int result = coinChange(coins, amount);
```

```
    if (result == -1)
        cout << "It is not possible.\n";
    else
        cout << "minimum coins needed: " << result << endl;

    return 0;
}
```

Output:

```
enter number of coins: 4
Enter coin values: 1 2 5 10
Enter amount: 111
minimum coins needed: 12
```

2.Maximum Ways to Give the Coin Change

```
#include <bits/stdc++.h>
using namespace std;

long long coinChange(vector<int> &coins, int amount) {
    int n = coins.size();

    vector<vector<long long>> dp(n + 1, vector<long long>(amount + 1, 0));
    for (int i = 0; i <= n; i++) {
        dp[i][0] = 1;
    }

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= amount; j++) {


            if (j >= coins[i - 1]) {

                dp[i][j] = dp[i - 1][j] + dp[i][j - coins[i - 1]];
```

```
            } else {
                dp[i][j] = dp[i - 1][j];
            }
        }
    }


    return dp[n][amount];
}

int main() {
    int n, amount;
    cout << "enter number of coins: ";
    cin >> n;

    vector<int> coins(n);
    cout << "Enter coin values: ";
    for (int i = 0; i < n; i++) {
        cin >> coins[i];
    }
    cout << "Enter target amount: ";
    cin >> amount;

    long long ways = coinChange(coins, amount);
    cout << "maximum number of ways: " << ways << endl;

    return 0;
}
```

```
enter number of coins: 3
Enter coin values: 1 2 3
Enter target amount: 5
maximum number of ways: 5
```