CoinChange #dynamicProgramming

```cpp
#include <bits/stdc++.h>
#include <iostream>

using namespace std;

const int numCoins = 3;
const int INF = 1000000009;
const int sum = 10;
int coins[numCoins] = {1, 3, 4};
int value [sum + 1];
int ready [sum + 1];

int sol (int sum) {
    if (sum < 0) return INF;
    if (sum == 0) return 0;
    if (ready[sum]) return value[sum];
    int best = INF;
    for (int i = 0; i < numCoins; ++i) {
        best = min (best, sol (sum - coins[i]) + 1);
    }
    ready[sum] = 1;
    value[sum] = best;
    return value[sum];
}

int main()
{
    cout << sol(10) << endl;
    /*
     * One thing that I've missed in the tutorial.
     * Suppose the coins I'm given are 2, 4, 5
     * now the sum I want  to form is 1.
     * Now that's not possible. That's why sol(1) will return me INF.
     * Now 'INF' means that there's no solution for forming that sum.
     */

    return 0;
}
```