Coin Change Permutation in C++

```cpp
#include <iostream>
#include <vector>

using namespace std;

int main() {
    vector<int> coins = {2, 3, 5, 6};
    int tar = 10;
    vector<int> dp(tar + 1, 0);
    dp[0] = 1; // Base case: 1 way to make amount 0 (using no coins)

    for (int amt = 1; amt <= tar; amt++) {
        for (int coin : coins) {
            if (coin <= amt) {
                int ramt = amt - coin;
                dp[amt] += dp[ramt];
            }
        }
    }

    cout << dp[tar] << endl; // Output the number of permutations to make the target amount

    return 0;
}
```

Dry Run:

Input:

coins = {2, 3, 5, 6}, target = 10

Initialization:

dp = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Loop Execution:

For amount amt = 1:

- coin = 2: No, as coin > amt.
- coin = 3: No, as coin > amt.
- coin = 5: No, as coin > amt.
- coin = 6: No, as coin > amt.

dp[1] = 0

For amount amt = 2:

- coin = 2: Yes, we can use one 2 to make 2. dp[2] += dp[0] (dp[0] is 1).
- coin = 3: No.
- coin = 5: No.
- coin = 6: No.

dp[2] = 1

For amount amt = 3:

- coin = 2: Yes, use one 2 and then add 1 way to make 1 (dp[1]).
- coin = 3: Yes, one 3 will form 3 (dp[0]).
- coin = 5: No.
- coin = 6: No.

dp[3] = 2

For amount amt = 4:

- coin = 2: Yes, use 2 and then form dp[2] ways.
- coin = 3: Yes, use 3 and then form dp[1] ways.
- coin = 5: No.
- coin = 6: No.

dp[4] = 3

For amount amt = 5:

- coin = 2: Yes, use 2 and form dp[3] ways.
- coin = 3: Yes, use 3 and form dp[2] ways.
- coin = 5: Yes, use 5 to make dp[0].

- coin = 6: No.

dp[5] = 4

For amount amt = 6:

- coin = 2: Yes, use 2 and form dp[4] ways.
- coin = 3: Yes, use 3 and form dp[3] ways.
- coin = 5: Yes, use 5 and form dp[1] ways.
- coin = 6: Yes, use 6 to make dp[0].

dp[6] = 5

For amount amt = 7:

- coin = 2: Yes, use 2 and form dp[5] ways.
- coin = 3: Yes, use 3 and form dp[4] ways.
- coin = 5: Yes, use 5 and form dp[2] ways.
- coin = 6: Yes, use 6 and form dp[1] ways.

dp[7] = 8

For amount amt = 8:

- coin = 2: Yes, use 2 and form dp[6] ways.
- coin = 3: Yes, use 3 and form dp[5] ways.
- coin = 5: Yes, use 5 and form dp[3] ways.
- coin = 6: Yes, use 6 and form dp[2] ways.

dp[8] = 12

For amount amt = 9:

- coin = 2: Yes, use 2 and form dp[7] ways.
- coin = 3: Yes, use 3 and form dp[6] ways.
- coin = 5: Yes, use 5 and form dp[4] ways.
- coin = 6: Yes, use 6 and form dp[3] ways.

dp[9] = 20

For amount amt = 10:

- coin = 2: Yes, use 2 and form dp[8] ways.
- coin = 3: Yes, use 3 and form dp[7] ways.
- coin = 5: Yes, use 5 and form dp[5] ways.
- coin = 6: Yes, use 6 and form dp[4] ways.

dp[10] = 33

Final Output:

dp[10] = 33

Output:-

```
33
```