



# The Coin Change Problem

by alanl

Problem

Submissions

Leaderboard

Discussions

Editorial

How many different ways can you make change for an amount, given a list of coins? In this problem, *your code* will need to efficiently compute the answer.

## Task

Write a program that, given

- The amount  $N$  to make change for and the number of types  $M$  of infinitely available coins
- A list of  $M$  coins -  $C = \{C_1, C_2, C_3, \dots, C_M\}$

Prints out how many different ways you can make change from the coins to STDOUT.

## The problem can be formally stated:

Given a value  $N$ , if we want to make change for  $N$  cents, and we have infinite supply of each of  $C = \{C_1, C_2, \dots, C_M\}$  valued coins, how many ways can we make the change? The order of coins doesn't matter.

## Solving the overlapping subproblems using dynamic programming

You can solve this problem recursively, but not all the tests will pass unless you optimise your solution to eliminate the [overlapping subproblems](#) using a [dynamic programming solution](#)

Or more specifically;

- If you can think of a way to store the checked solutions, then this store can be used to avoid checking the same solution again and again.

## Input Format

First line will contain 2 integer N and M respectively.

Second line contain M integer that represent list of distinct coins that are available in infinite amount.

## Constraints

- $1 \leq C_i \leq 50$
- $1 \leq N \leq 250$
- $1 \leq M \leq 50$
- The list of coins will contain distinct integers.

## Output Format

One integer which is the number of ways in which we can get a sum of N from the given infinite supply of M types of coins.

## Sample Input

```
4 3
1 2 3
```

**Sample Output**

4

**Sample Input #02**

```
10 4
2 5 3 6
```

**Sample Output #02**

5

**Explanation**

- *Example 1:* For  $N = 4$  and  $C = \{1, 2, 3\}$  there are four solutions:  $\{1, 1, 1, 1\}$ ,  $\{1, 1, 2\}$ ,  $\{2, 2\}$ ,  $\{1, 3\}$
- *Example 2:* For  $N = 10$  and  $C = \{2, 5, 3, 6\}$  there are five solutions:  $\{2, 2, 2, 2, 2\}$ ,  $\{2, 2, 3, 3\}$ ,  $\{2, 2, 6\}$ ,  $\{2, 3, 5\}$ ,  $\{5, 5\}$ .

**Hints**

- Think about the degenerate cases:
  - How many ways can you give change for 0 cents?
  - How many ways can you give change for >0 cents, if you have no coins?
- If you are having trouble defining your solutions store, then think about it in terms of the base case ( $n = 0$ )
- For help on reading from STDIN, see the [HackerRank environment help page](#) under the "Sample Problem Statement" section.

**Environment and Samples**

Environment for the programming languages   **Sample Problem Statement**   Writing state information to a file

f t in


Submissions: 22096

Max Score: 60

Difficulty: Hard

Rate This Challenge:

☆☆☆☆☆

[More](#)Current Buffer (saved locally, editable)  

C++



```
1  /*
2  Exapmle that time-out
3  -----
4  Input:
5  166 23
6  5 37 8 39 33 17 22 32 13 7 10 35 40 2 43 49 46 19 41 1 12 11 28
7  Output:
8  96190959
9
10 Idea to improve speed:
11 - Sort coins
12 - then break out of the for loop when new_change >= n
13 */
14
15 #include <cmath>
16 #include <cstdio>
17 #include <vector>
18 #include <iostream>
19 #include <algorithm>
```

```
20 using namespace std;
21
22 void coin_change(int n, int l, int m, int c[], int change, int &change_count) {
23     if (l == m) return;
24
25     for (int i=l; i<m; i++) {
26         int new_change = change + c[i];
27         if (new_change == n) {
28             change_count++;
29         }
30
31         if (new_change < n) {
32             coin_change(n, i, m, c, new_change, change_count);
33         }
34     }
35 }
36
37 int main() {
38     int N;
39     int M;
40
41     cin >> N;
42     cin >> M;
43
44     int C[M];
45
46     for (int i=0; i<M; i++) {
47         cin >> C[i];
48     }
49
50     int change_count = 0;
51     coin_change(N, 0, M, C, 0, change_count);
52     cout << change_count << endl;
53
54     return 0;
55 }
56
```

Line: 2 Col: 13

 Upload Code as File☐ Test against custom input

Run Code

Submit Code

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)