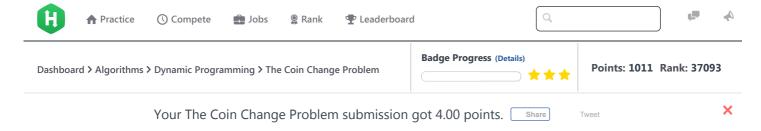
2/21/2018 HackerRank



The Coin Change Problem



Problem	Submissions	Leaderboard	Discussions	Editorial 🔒		
---------	-------------	-------------	-------------	-------------	--	--

In this problem you will be given a list of coin denominations and a target amount. Determine the number of ways the target amount can be arrived at using the denominations available.

Input Format

- 2 space-separated integers
 - n and m: target amount, number of denominations
 - $1 \le n \le 250$
 - $1 \le m \le 50$
- m space-separated integers
 - c_i : unique coin denominations
 - $1 \le c_i \le 50$

Hints

- Solve overlapping subproblems using Dynamic Programming (DP):
 You can solve this problem recursively but will not pass all the test cases without optimizing to eliminate the overlapping subproblems. Think of a way to store and reference previously computed solutions to avoid solving the same subproblem multiple times.
- Consider the degenerate cases:
 - How many ways can you make change for **0** cents?
 - How many ways can you make change for > 0 cents if you have no coins?
- ullet If you're having trouble defining your solutions store, then think about it in terms of the base case (n=0).
- $\bullet~$ The answer may be larger than a ${\bf 32}\text{-bit}$ integer.

Output Format

Print a long integer denoting the number of ways we can get a sum of n from the given infinite supply of M types of coins.

Sample Input 0

4 3

1 2 3

Sample Output 0

4

Explanation 0

2/21/2018 HackerRank

Given coins of denominations [1, 2, 3] and a target amount of 4, the following 4 sets of coins meet the goal: [1, 1, 1, 1], [1, 1, 2], [2, 2] and [1, 3].

Sample Input 1

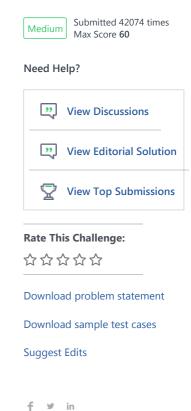
```
10 4
2 5 3 6
```

Sample Output 1

5

Explanation 1

Given coins of denominations [2, 5, 3, 6] and a target amount of $\mathbf{10}$, the following $\mathbf{5}$ sets of coins meet the goal: [2, 2, 2, 2], [2, 2, 3, 3], [2, 2, 6], [2, 3, 5], [5, 5]



```
Current Buffer (saved locally, editable) \ \mathscr{V} \ \mathfrak{O}
                                                                                          C++14
                                                                                                                             *
 1 ▼ #include <iostream>
 2 #include <vector>
   #include <iterator>
 4 #include <algorithm>
   #include <numeric>
    #include <functional>
    using namespace std;
    typedef unsigned long long int uint64;
10 inline uint64 CoinChangeProblem(int &targetAmt, int &denomination, vector<int> &coins)
11 ▼ {
12
        vector<uint64> targetVec(targetAmt+1);
13
        sort(coins.begin(), coins.end());
14
        for(auto idx = 0; idx<denomination; ++idx)</pre>
15 ▼
16
             if(idx == 0)
                 targetVec[idx] = coins[idx];
17
18
             for(auto i=0; i< targetAmt+1; ++i)</pre>
19 •
                 int amt = i;
20
21 •
                 if(amt>=coins[idx])
                     targetVec[amt] +=targetVec[amt-coins[idx]];
22 ▼
```

2/21/2018 HackerRank

```
23
24
25
         for(auto &it: targetVec)
26
             cout<<it<<" ";
27
28 ▼
        return targetVec[targetAmt];
29 }
30
31 int main()
32 ▼ {
33
        int targetAmt, denomination;
        cin>>targetAmt;
                                      // target amount,
34
35
        cin>>denomination;
                                      // number of denominations
36
37
        vector<int> coins; coins.reserve(denomination);
38
        copy_n(istream_iterator<int>(cin), denomination, back_inserter(coins));
39
        cout<<CoinChangeProblem(targetAmt, denomination, coins)<<endl;</pre>
40
41
        return 0;
42
    }
43
                                                                                                                 Line: 26 Col: 5
                                                                                                      Run Code
                       Test against custom input
                                                                                                                   Submit Code
1 Upload Code as File
 Testcase 0 ×
               Testcase 1 🗙
 Your code did not pass this test case.
 Input (stdin)
  10 4
  2 5 3 6
 Your Output (stdout)
  2 0 2 2 2 4 6 4 8 8 10 10
```

 $Contest\ Calendar |Blog|Scoring|Environment|FAQ|About\ Us|Support|Careers|Terms\ Of\ Service|Privacy\ Policy|Request\ a\ Feature$

Expected Output

Compiler Message Wrong Answer

5