You are working at the cash counter at a fun-fair, and you have different types of coins available to you in infinite quantities. The value of each coin is already given. Can you determine the number of ways of making change for a particular number of units using the given types of coins?

For example, if you have 4 types of coins, and the value of each type is given as $8, 3, 1, 2$ respectively, you can make change for 3 units in three ways: $\{1, 1, 1\}$, $\{1, 2\}$, and $\{3\}$.

Function Description

Complete the *getWays* function in the editor below. It must return an integer denoting the number of ways to make change.

getWays has the following parameter(s):

- *n*: an integer, the amount to make change for
- *c*: an array of integers representing available denominations

Input Format

The first line contains two space-separated integers describing the respective values of n and m, where:
n is the number of units
m is the number of coin types
The second line contains m space-separated integers describing the respective values of each coin type
: $c = [c[0], c[1] \ldots c[m-1]]$ (the list of distinct coins available in infinite amounts).

Constraints

- $1 \leq c[i] \leq 50$
- $1 \leq n \leq 250$
- $1 \leq m \leq 50$
- Each $c[i]$ is guaranteed to be distinct.

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? * If you're having trouble defining your solutions store, then think about it in terms of the base case $(n = 0)$. - The answer may be larger than a 32-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

There are four ways to make change for $n = 4$ using coins with values given by $C = [1, 2, 3]$:

1. $\{1, 1, 1, 1\}$
2. $\{1, 1, 2\}$
3. $\{2, 2\}$

4. $\{1, 3\}$

Thus, we print 4 as our answer.

Sample Input 1

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

Sample Output 1

```
5
```

Explanation 1

There are five ways to make change for $n = 10$ units using coins with values given by $C = [2, 5, 3, 6]$:

1. $\{2, 2, 2, 2, 2\}$
2. $\{2, 2, 3, 3\}$
3. $\{2, 2, 6\}$
4. $\{2, 3, 5\}$
5. $\{5, 5\}$

Thus, we print 5 as our answer.