**Coin Change**

**Medium**

Given an integer array **coins[ ]**of size**N**representing different denominations of currency and an integer **sum**, find the number of ways you can make **sum** by using different combinations from coins[ ].    
**Note:** Assume that you have an infinite supply of each type of coin. And you can use any coin as many times as you want.

**Example 1:**

**Input:**

N = 3, sum = 4

coins = {1,2,3}

**Output:** 4

**Explanation**: Four Possible ways are: {1,1,1,1},{1,1,2},{2,2},{1,3}.

**Example 2:**

**Input**:

N = 4, Sum = 10

coins = {2,5,3,6}

**Output:** 5

**Explanation**: Five Possible ways are: {2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5} and {5,5}.

**Expected Time Complexity:**O(sum\*N)  
**Expected Auxiliary Space:**O(sum)

**Constraints:**  
1 <= sum, N, coins[i] <= 103

**Company Tags**

[**Paytm**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Paytm) [**Flipkart**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Flipkart) [**Morgan Stanley**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Morgan%20Stanley) [**Accolite**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Accolite) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**OYO Rooms**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=OYO%20Rooms) [**Samsung**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Samsung) [**Snapdeal**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Snapdeal)

**Topic Tags**

[**Arrays**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Arrays) [**Dynamic Programming**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Dynamic%20Programming) [**Data Structures**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Data%20Structures) [**Algorithm**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Algorithms)

//{ Driver Code Starts

// Initial Template for Java

import java.io.\*;

import java.util.\*;

class CodingMaxima {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while (t-- > 0) {

int sum = sc.nextInt();

int N = sc.nextInt();

int coins[] = new int[N];

for (int i = 0; i < N; i++) coins[i] = sc.nextInt();

Solution ob = new Solution();

System.out.println(ob.count(coins, N, sum));

}

}

}

// } Driver Code Ends

class Solution {

public long count(int coins[], int N, int sum) {

// code here.

long[][] arr = new long[N + 1][sum + 1];

for(int i = 0; i < N + 1; i++)

arr[i][0] = 1;

for(int i = 1; i < N + 1; i++)

for(int j = 0; j < sum + 1; j++)

if(coins[i - 1] > j)

arr[i][j] = arr[i - 1][j];

else

arr[i][j] = arr[i][j - coins[i - 1]] + arr[i - 1][j];

return arr[N][sum];

}

}