**Practice Problem: Subset Sum**

**Exact Change 2 (change2)**

Whenever the UCF Programming Team travels to World Finals, Glenn likes having the exact amount of money necessary for any purchase, so that he doesn’t have to count and receive change. Of course, most countries don’t have many different denominations of coins, so Glenn creates different “packages” with him, each with some particular amount of money, in cents. Given the price of a particular item, Glenn would like to know if he can pay the exact amount necessary with some combination of his packages.

**The Problem:**

Given a list of positive integers and a target value, T, determine if some subset of the integers given adds up to T exactly.

**The Input:**

The first input line contains a single positive integer, *n (1 < n ≤ 100)*, indicating the number of purchases to evaluate. Each of the *n* input sets follows. The first line of each input set contains only an integer, *c (1 ≤ c < 20)*, representing the number of different packages of coin for that input set. The following line contains exactly *c* positive integers, each separated by a single space, representing the value of each of the *c* packages in cents. The sum of these *c* integers is guaranteed not to exceed 2000000000. Note that the package values are not necessarily distinct, i.e., there may be multiple packages with the same value. The last line of each input set will contain a single positive integer, *T*, the price of the item Glenn wants to purchase, in cents, guaranteed not to exceed 109.

**The Output:**

For each set of packages, first output “Item #i: ” where *i* is the input data set number, starting with 1. If some subset of the package values equals the gien target value, follow this with the string, “YES”. Otherwise, follow this with the string “NO”. Note that a package value can be used at most once in a subset unless there are multiple packages with that value (if there are *m* occurrences of a package value, up to *m* occurrences of that value can be used in a subset). Follow the format illustrated in Sample Output.

**Sample Input: Sample Output:**

3 Item #1: NO

6 Item #2: YES

12 8 1 2 4 100 Item #3: YES

28

3

1 2 3

4

6

3 1 3 2 3 88

91