

Homework 5, CSE 232

Due October 1st Note: On most of the problem sets through the semester, I'll put a horizontal line with "Optional" under it. Any problems below this section are encouraged - I think they're interesting and will help you learn the subject - but not necessary to complete in order to get credit for the homework.

Problem 1

This week's problem is [GCJ Qualification 2011 - Candy Splitting](#).

Here's the problem statement:

Sean and Patrick are brothers who just got a nice bag of candy from their parents. Each piece of candy has some positive integer value, and the children want to divide the candy between them. First, Sean will split the candy into two piles, and choose one to give to Patrick. Then Patrick will try to calculate the value of each pile, where the value of a pile is the sum of the values of all pieces of candy in that pile; if he decides the piles don't have equal value, he will start crying.

Unfortunately, Patrick is very young and doesn't know how to add properly. He almost knows how to add numbers in binary; but when he adds two 1s together, he always forgets to carry the remainder to the next bit. For example, if he wants to sum 12 (1100 in binary) and 5 (101 in binary), he will add the two rightmost bits correctly, but in the third bit he will forget to carry the remainder to the next bit:

$$1100 + 0101 \text{ --- } 1001$$

So after adding the last bit without the carry from the third bit, the final result is 9 (1001 in binary). Here are some other examples of Patrick's math skills:

$$5 + 4 = 1$$

$$7 + 9 = 14$$

$$50 + 10 = 56$$

Sean is very good at adding, and he wants to take as much value as he can without causing his little brother to cry. If it's possible, he will split the bag of candy into two non-empty piles such that Patrick thinks that both have the same value. Given the values of all pieces of candy in the bag, we would like to know if this is possible; and, if it's possible, determine the maximum possible value of Sean's pile.

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case is described in two lines. The first line contains a single integer N , denoting the number of candies in the bag. The next line contains the N integers C_i separated by single spaces, which denote the value of each piece of candy in the bag.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1). If it is impossible for Sean to keep Patrick from crying, y should be the word "NO". Otherwise, y should be the value of the pile of candies that Sean will keep.

Limits

$$1 \leq T \leq 100.$$

$$1 \leq C_i \leq 106.$$

Small dataset

$$2 \leq N \leq 15.$$

Large dataset

$$2 \leq N \leq 1000.$$

a) *Using the way in which Patrick adds numbers, write a function that takes a list of candy values and returns how much Patrick thinks all the candies are worth together.*

b) *What are the maximum number of possible ways Sean can divide the candies into two piles for the Small and the Large cases? Write a complete search solution that tests each possible division for whether Patrick is satisfied and finds the maximum value for Sean's pile.*

c) *We can calculate whether two numbers are equal using no direct comparison operators (e.g., “==”, “!=”, etc.) but instead with only a single binary operator. Write a function that, given, Patrick's estimate of the values of two piles of candy, returns a Boolean describing whether he thinks they're equal.*

d) *Given your answers to the above questions, how does Patrick's satisfaction depend on how Sean divides the candies into two piles? Use this result to write a linear-time solution to the problem that correctly solves the Large input.*

Optional No optional work this week.