**Patching**

There is a popular video game that people like playing involving numbers, but there is one glaring issue: not all numbers are able to be made! They would like you to add patches such that any number in the range [1, **S**] (inclusive) can be formed by the sum of some elements in the array. Do you think you can add the minimum number of patches?

**Input:** The first line of input denotes **T**, the number of test cases. The first line of each test case contains **N**, the number of elements and **S**, the upper bound of the range. In the second line, there will be **N** space-separated integers, they will be given in sorted order.

**Output:** You will first output “CASE #(case number): “ followed by the minimum number of patches required. If there are no patches required, then output 0.

**Example Input:**

3

2 6

1 3

3 20

1 5 10

3 5

1 2 2

**Example Output:**

CASE #1: 1

CASE #2: 2

CASE #3: 0

**Explanation:** For case #1: the beginning list of numbers is [1, 3], and the upper bound is 6. The current combinations we can make are [1], [3], and [1, 3], which add up to 1, 3, 4 respectively. If we add the integer 2 to the list, then the new combinations are

[1], [2], [3], [1, 3], [2, 3], [1, 2, 3], which all sum to be 1, 2, 3, 4, 5, 6 respectively. This now covers the range [1, 6] inclusive, so we only needed one patch.

Using the same logic, the patch for case #2, would be the numbers 2, and 4.

For case #3, there is no patch necessary since the number between 1 and 5 inclusive can already be made with the following list.