Round 1C 2008

A. Text Messaging Outrage

B. Ugly Numbers

C. Increasing Speed Limits

Contest Analysis

Questions asked 4

Problem A. Text Messaging Outrage

Small input 5 points	Solve A-small
Large input 10 points	Solve A-large

The story

Professor Loony, a dear friend of mine, stormed into my office. His face was red and he looked very angry. The first thing that came out of his mouth was "Damn those phone manufacturers. I was trying to send a text message, and it took me more than ten minutes to type a one-line message." I tried to calm him down. "But what is wrong? Why did it take you so long?" He continued, "Don't you see?! Their placement of the letters is so messed up? Why is 's' the 4th letter on its key? and 'e'? Why is it not the first letter on its key? I have to press '7' FOUR times to type an 's'? This is lunacy!"

"Calm down, my friend," I said, "This scheme has been in use for so long, even before text messaging was invented. They had to keep it that way."

"That's not an excuse," his face growing redder and redder. "It is time to change all this. It was a stupid idea to start with. And while we are at it, how come they only put letters on 8 keys? Why not use all 12? And why do they have to be consecutive?"

"Umm... I... don't... know," I replied.

"Ok, that's it. Those people are clearly incompetent. I am sure someone can come up with a better scheme."

He was one of *those* people, I could see. People who complain about the problem, but never actually try to solve it.

In this problem, you are required to come up with the best letter placement of keys to minimize the number of key presses required to type a message. You will be given the number of keys, the maximum number of letters we can put on every key, the total number of letters in the alphabet, and the frequency of every letter in the message. Letters can be placed anywhere on the keys and in any order. Each letter can only appear on one key. Also, the alphabet can have more than 26 letters (it is not English).

For reference, the current phone keypad looks like this

```
key 2: abc
key 3: def
key 4: ghi
key 5: jkl
key 6: mno
key 7: pqrs
key 8: tuv
key 9: wxyz
```

The first press of a key types the first letter. Each subsequent press advances to the next letter. For example, to type the word "snow", you need to press "7" four times, followed by "6" twice, followed by "6" three times, followed by "9" once. The total number of key presses is 10.

Input

The first line in the input file contains the number of test cases **N**. This is followed by **N** cases. Each case consists of two lines. On the first line we have the maximum number of letters to place on a key (**P**), the number of keys available (**K**) and the number of letters in our alphabet (**L**) all separated by single spaces. The second line has **L** non-negative integers. Each number represents the frequency of a certain letter. The first number is how many times the first letter is used, the second number is how many times the second letter is used, and so on.

Output

For each case, you should output the following

```
Case #x: [minimum number of keypad presses]
```

indicating the number of keypad presses to type the message for the optimal layout.

Limits

P * **K** ≥ **L**
0 ≤ The frequency of each letter ≤ 1 000 000

Small dataset

1 ≤ **N** ≤ 10
1 ≤ **P** ≤ 10
1 ≤ **K** ≤ 12
1 ≤ **L** ≤ 100

Large dataset

1 ≤ **N** ≤ 100
1 ≤ **P** ≤ 1 000
1 ≤ **K** ≤ 1 000
1 ≤ **L** ≤ 1 000

Sample

Input

```
2
3 2 6
8 2 5 2 4 9
3 9 26
1 1 1 100 100 1 1 1 1 1 1 1 1 1 1 1 10 11 11 11 11 1 1
1 100
```

Output

```
Case #1: 47
Case #2: 397
```