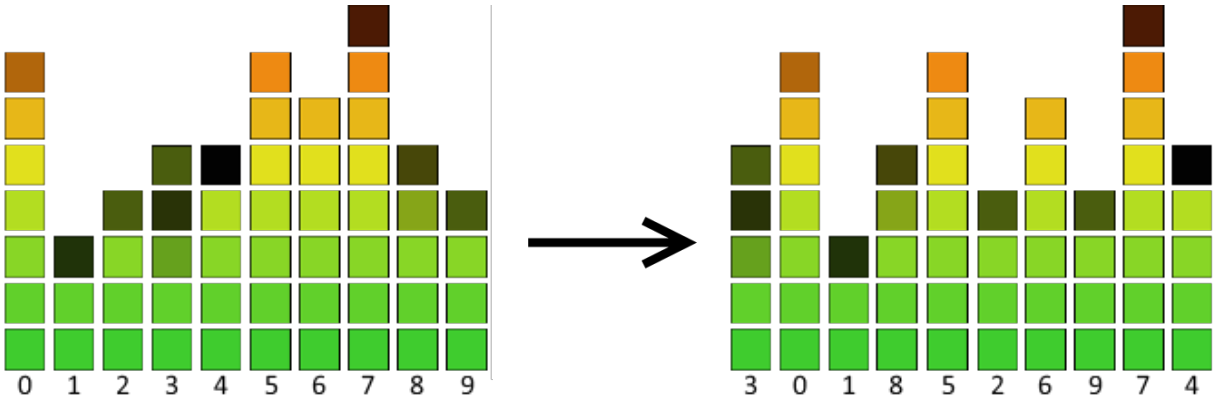
Disc Jockey

Filename: `disc`

Gerald's brother, Charlie, recently had a birthday party. Since all of Charlie's friends know that he wants to become a D.J. when he grows up, they gave him many cool new gifts related to music. However, the best gift was a new stereo which had a fancy graphic visualizer on the front. A graphic visualizer is a feature found in electronic music devices which generates animated imagery synchronized to music being played. Charlie's visualizer has *n* different columns of squares which represent the amplitude of a particular frequency.



Gerald is very jealous of his little brother's new stereo and can't stand to listen to the music that his brother plays through its speakers, so he has devised a plot to solve his problem. He has a theory for how sound works, and it is this: All sounds can be assigned a value *s*. To calculate *s*, you must sum all the absolute difference in heights between each pair of adjacent columns. For example, in the left image above, column 0 and column 1 are 4 apart, column 1 and column 2 are 1 apart, and so on. If you add all these differences you would get a value of $s = 15$. Gerald also believes that if a sound wave's columns are rearranged in such a way that *s* is as large as possible and then is played at the same time as the original sound, the two will cancel out and no sound will be heard. There may be more than one valid way to do this. If you are to rearrange the columns in the original wave to the order on the right, you can achieve a maximum $s = 24$. Note that an entire column must be rearranged when done so and not just portions of it.

The Problem:

Help Gerald find the maximum value of *s* for a given sound wave represented by *n* columns when he is able to swap around any columns. This way he can cancel out all of his brother's music and never be bothered again!

The Input:

The first line will contain a single, positive integer, *t*, representing the number of different wave forms that need to be processed. The next *t* lines will start with an integer, *n* ($1 \leq n \leq 10$), followed by *n* integers, c_0 through c_{n-1} ($0 \leq c_i \leq 10^8$), representing the height of each column in a wave representation.

The Output:

For each wave, output a single line of the form "`Wave #i: s`" where *i* represents the number of the wave which you are processing (starting from 1), and *s* represents the maximum *s* value of that wave.

Sample Input:

```
3
3 1 2 3
5 9 2 5 3 1
10 7 3 4 5 5 7 6 8 5 4
```

Sample Output:

```
Wave #1: 3
Wave #2: 21
Wave #3: 24
```