04. ArrayLists/Vectors 1

- Comparing arrays and ArrayLists/Vectors
  - Similarity
  - Difference
- How to create an ArrayList?
- How to access the elements of an ArrayList?
- How to add/remove elements to/from an ArrayList?

Exercise 1: Longest wave

Exercise 2: Largest divisible subsequence

Exercise 3: Longest increasing subsequence

Exercise 4: Consecutive subarrays

Exercise 5: Not last (Jan 2017)

Exercise 6: Crosswords (Dec 2014)

Longest wave

Given an array A of integers, a wave is a subsequence $A_{i_1}, A_{i_2}, \ldots, A_{i_k}$ so that $i_1 < i_2 < \cdots < i_k$ but $A_{i_1} < A_{i_2}$, $A_{i_2} > A_{i_3}$, $A_{i_3} < A_{i_4}$, $A_{i_4} > A_{i_5}$, ….

Report the max length of any wave of array A.

Input (from terminal / stdin)

- The first line contains integer N, $1 \leq N \leq 1000$.
- The second line contains the N integers, all in the range [0, 1e6].

Output (to terminal / stdout)

- Report the max length of any wave.

Sample input
5
1 3 5 4 8

Sample output
4

The subsequence {1, 5, 4, 8} and the subsequence {3, 5, 4, 8} both have max length 4.

Largest divisible subsequence

Given an array A of distinct positive integers in ascending order, report the max size of any subsequence of A so that for any two integers in this subsequence, the latter is a multiple of the former.

Input (from terminal / stdin)

- The first line contains integer N, $1 \leq N \leq 1000$.
- The next line contains the N integers, all in the range [1, 1e6].

Output (to terminal / stdout)

- Report the max length of any subsequence satisfying the restriction and then the subsequence. If there are multiple subsequences exist, print the smallest one in lexicographic order.

Sample input
5
1 2 5 6 8

Sample output
3
1 2 6

The subsequences {1, 2, 6} and {1, 2, 8} both have the max length 3, but {1, 2, 6} is the smallest.

Longest increasing subsequence

Given an array A of integers, find the max length of any subsequence that is increasing.

Input (from terminal / stdin)

- The first line contains integer N, $1 \leq N \leq 1000$.
- The next line contains the N integers, all in the range [0, 1e6].

Output (to terminal / stdout)

- Report the max length of any increasing subsequence and the subsequence. If there are multiple such subsequences, you can print any one.

Sample input
6
1 3 6 5 8 2

Sample output
4
1 3 6 8

Both the increasing subsequences {1, 3, 6, 8}, {1, 3, 5, 8} have the max length 4.

Consecutive subarrays

Given an array A of integers, remove the subarrays of increasing consecutive integers of length at least K. Remove such subarrays from the beginning of the array. Note that removing a subarray may yield some previously hidden subarrays of consecutive integers.

Input (from terminal / stdin)

- The first line contains integers N and K, $1 \leq K \leq N \leq 1e5$.
- The second line contains the N integers, all in the range $[0, 1e6]$.

Output (to terminal / stdout)

- Report the remaining integers after removing the continuous subarrays of length $\geq K$. If there is no integer remaining, print "NONE".

Sample input 1
5 3
1 3 4 5 2

Sample output 1
1 2

Sample input 2
8 3
1 3 4 5 2 3 4 2

Sample output 2
2

After removing subarray 3 4 5, the array becomes 1 2 3 4 2. Then after removing the subarray 1 2 3 4, only integer 2 is left.

Not last (Jan 2017)

Farmer John owns 7 dairy cows: Bessie, Elsie, Daisy, Gertie, Annabelle, Maggie, and Henrietta. He milks them every day and keeps detailed records on the amount of milk provided by each cow during each milking session. Not surprisingly, Farmer John highly prizes cows that provide large amounts of milk.

Cows, being lazy creatures, don't necessarily want to be responsible for producing too much milk. If it were up to them, they would each be perfectly content to be the lowest-producing cow in the entire herd. However, they keep hearing Farmer John mentioning the phrase "farm to table" with his human friends, and while they don't quite understand what this means, they have a suspicion that it may not be the best idea to be the cow producing the least amount of milk. Instead, they figure it's safer to be in the position of producing the second-smallest amount of milk in the herd. Please help the cows figure out which of them currently occupies this desirable position.

INPUT FORMAT (file notlast.in):

- The input file for this task starts with a line containing the integer N ($1 \leq N \leq 100$), giving the number of entries in Farmer John's milking log.
- Each of the N following lines contains the name of a cow (one of the seven above) followed by a positive integer (at most 100), indicating the amount of milk produced by the cow during one of its milking sessions.

  Any cow that does not appear in the log at all is assumed to have produced no milk.

OUTPUT FORMAT (file notlast.out):

On a single line of output, please print the name of the cow that produces the second-smallest amount of milk. More precisely, if M is the minimum total amount of milk produced by any cow, please output the name of the cow whose total production is minimal among all cows that produce more than M units of milk. If several cows tie for this designation, or if no cow has this designation (i.e., if all cows have production equal to M), please output the word "Tie". Don't forget to add a newline character at the end of your line of output. Note that $M = 0$ if one of the seven cows is completely absent from the milking log, since this cow would have produced no milk.

SAMPLE INPUT:

```
10
Bessie 1
Maggie 13
Elsie 3
Elsie 4
Henrietta 4
Gertie 12
Daisy 7
Annabelle 10
Bessie 6
Henrietta 5
```

SAMPLE OUTPUT:

```
Henrietta
```

In this example, Bessie, Elsie, and Daisy all tie for the minimum by each producing 7 units of milk. The next-largest production, 9 units, is due to Henrietta.

Crosswords (Dec 2014)

Like all cows, Bessie the cow likes to solve crossword puzzles. Unfortunately, her sister Elsie has spilled milk all over her book of crosswords, smearing the text and making it difficult for her to see where each clue begins. It's your job to help Bessie out and recover the clue numbering!

An unlabeled crossword is given to you as an N by M grid ($3 \leq N \leq 50$, $3 \leq M \leq 50$). Some cells will be clear (typically colored white) and some cells will be blocked (typically colored black). Given this layout, clue numbering is a simple process which follows two logical steps:

Step 1: We determine if a cell begins a horizontal or vertical clue. If a cell begins a horizontal clue, it must be clear, its neighboring cell to the left must be blocked or outside the crossword grid, and the two cells on its right must be clear (that is, a horizontal clue can only represent a word of 3 or more characters). The rules for a cell beginning a vertical clue are analogous: the cell above must be blocked or outside the grid, and the two cells below must be clear.

Step 2: We assign a number to each cell that begins a clue. Cells are assigned numbers sequentially starting with 1 in the same order that you would read a book; cells in the top row are assigned numbers from left to right, then the second row, etc. Only cells beginning a clue are assigned numbers.

For example, consider the grid, where '.' indicates a clear cell and '#' a blocked cell.

```
...
#..
...
..#
.##
```

Cells that can begin a horizontal or vertical clue are marked with ! below:

```
!!!
#..
!..
..#
.##
```

If we assign numbers to these cells, we get the following;

```
123
#..
4..
..#
.##
```

Note that crossword described in the input data may not satisfy constraints typically seen in published crosswords. For example, some clear cells may not be part of any clue.

INPUT: (file crosswords.in)

- The first line of input contains N and M separated by a space.
- The next N lines of input each describe a row of the grid. Each contains M characters, which are either '.' (a clear cell) or '#' (a blocked cell).

OUTPUT: (file crosswords.out)

- On the first line of output, print the number of clues.

- On the each remaining line, print the row and column giving the position of a single clue (ordered as described above). The top left cell has position (1, 1). The bottom right cell has position (N, M).

SAMPLE INPUT:

```
5 3
...
#..
...
..#
.##
```

SAMPLE OUTPUT:

```
4
1 1
1 2
1 3
3 1
```