01. Loops

Loops
- for-loops
- while-loops
- do-while loops
- nested loops
- break
- continue

Exercise 1: Moobuzz

Exercise 2: Largest so far

Exercise 3: Lattice points in ellipses

Exercise 4: Right triangles

Exercise 5: Mix milk (Dec 2018)

Exercise 6: Cow jog (Dec 2014)

Moobuzz

Given integer N, create the sequence of N elements

1, 2, moo, 4, buzz, moo, 7, 8, moo, buzz, 11, moo, 13, 14, Moobuzz, 16, 17, …

That is, replace an integer that is only a multiple of 3 with “moo”, an integer that is only a multiple of 5 with “buzz”, and a multiple of both 3 and 5 with “Moobuzz”.

Input (from terminal / stdin)

- The only line contains integer N, $1 \leq N \leq 1000$.

Output (to terminal / stdout)

- Display the sequence of N elements in one line (separated by white spaces).

Sample input
```
16
```

Sample output
```
1 2 moo 4 buzz moo 7 8 moo buzz 11 moo 13 14 Moobuzz 16
```

Largest so far

Given an array of integers, count the number of integers that are larger than all integers to the left.

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, 1000].

Output (to terminal / stdout)

- Report the number of integers that are larger than all integers to the left.

Sample input

```
4
3 1 3 5
```

Sample output

```
2
```

Lattice points in ellipses

Given two ellipses

$$\frac{(x-h_1)^2}{a_1^2}+\frac{(y-k_1)^2}{b_1^2}=1,\quad \frac{(x-h_2)^2}{a_2^2}+\frac{(y-k_2)^2}{b_2^2}=1$$

report the number of lattice points inside the overlap of them.

Input (from terminal / stdin)

- The first line contains the four integers, a_1, b_1, h_1, k_1.
- The second line contains the four integers, a_2, b_2, h_2, k_2.
- Assume the integers are in the range [1, 100].

Output (to terminal / stdout)

- Report the lattice points in the interior of the two ellipses.

Sample input
```
1 2 0 0
2 1 0 0
```

Sample output
```
5
```

Right triangles

There are N distinct points $P_i = (x_i, y_i)$ $(1 \leq i \leq N)$ in the first quadrant and M distinct points $Q_j = (x_j, 0)$ $(1 \leq j \leq M)$ on the x-axis. Report the number of right triangles that can be formed using two points $Q_j = (x_j, 0)$, $Q_k = (x_k, 0)$ $(j \neq k)$ and one point $P_i = (x_i, y_i)$.

Input (from terminal / stdin)

- The first line contains integers N, and M, $1 \leq N, M \leq 100$.
- Each of the next N lines contains the coordinates of one point $P_i = (x_i, y_i)$ where all coordinates are in the range [1, 100].
- The next line contains the x-coordinates of the M points $Q_j = (x_j, 0)$ where all the coordinates x_j are in the range [1, 100].

Output (to terminal / stdout)

- Report the number of right triangles that can formed.

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

Sample output
```
4
```

Mix milk (Dec 2018)

Farming is competitive business – particularly milk production. Farmer John figures that if he doesn't innovate in his milk production methods, his dairy business could get creamed!

Fortunately, Farmer John has a good idea. His three prize dairy cows Bessie, Elsie, and Mildred each produce milk with a slightly different taste, and he plans to mix these together to get the perfect blend of flavors.

To mix the three different milks, he takes three buckets containing milk from the three cows. The buckets may have different sizes, and may not be completely full. He then pours bucket 1 into bucket 2, then bucket 2 into bucket 3, then bucket 3 into bucket 1, then bucket 1 into bucket 2, and so on in a cyclic fashion, for a total of 100 pour operations (so the 100th pour would be from bucket 1 into bucket 2). When Farmer John pours from bucket a into bucket b, he pours as much milk as possible until either bucket a becomes empty or bucket b becomes full.

Please tell Farmer John how much milk will be in each bucket after he finishes all 100 pours.

INPUT FORMAT (file mixmilk.in):
The first line of the input file contains two space-separated integers: the capacity c_1 of the first bucket, and the amount of milk m_1 in the first bucket. Both c_1 and m_1 are positive and at most 1 billion, with $c_1 \geq m_1$. The second and third lines are similar, containing capacities and milk amounts for the second and third buckets.

OUTPUT FORMAT (file mixmilk.out):
Please print three lines of output, giving the final amount of milk in each bucket, after 100 pour operations.

SAMPLE INPUT:
```
10 3
11 4
12 5
```

SAMPLE OUTPUT:
```
0
10
2
```

In this example, the milk in each bucket is as follows during the sequence of pours:

```
Initial State: 3  4  5
1. Pour 1->2:  0  7  5
2. Pour 2->3:  0  0  12
3. Pour 3->1:  10 0  2
4. Pour 1->2:  0  10 2
5. Pour 2->3:  0  0  12
(The last three states then repeat in a cycle ...)
```

Use a small number of total buckets, but he is not sure how many exactly. Please help him out!

Cow jog (Dec 2014)

The cows are out exercising their hooves again! There are N cows jogging on an infinitely-long single-lane track ($1 \leq N \leq 100{,}000$). Each cow starts at a distinct position on the track, and some cows jog at different speeds.

With only one lane in the track, cows cannot pass each other. When a faster cow catches up to another cow, she has to slow down to avoid running into the other cow, becoming part of the same running group.

Eventually, no more cows will run into each other. Farmer John wonders how many groups will be left when this happens. Please help him compute this number.

INPUT: (file cowjog.in)

The first line of input contains the integer N.

The following N lines each contain the initial position and speed of a single cow. Position is a nonnegative integer and speed is a positive integer; both numbers are at most 1 billion. All cows start at distinct positions, and these will be given in increasing order in the input.

OUTPUT: (file cowjog.out)

A single integer indicating how many groups remain.

SAMPLE INPUT:
```
5
0 1
1 2
2 3
3 2
6 1
```

SAMPLE OUTPUT:
```
2
```