

G Merry-Go-Round

While hundreds of joyful children ride a merry-go-round, you have to analyze it. The merry-go-round has N horses, each with some number of children riding it. As the merry-go-round rotates, all the kids get off the first horse. One kid from the first horse will get onto the second, one onto third, and so on until there are no kids left on the first horse. Generally, if K children get off horse a , 1 child will get onto each of the K horses after a . If there are more than N kids on the horse, then one kid will get onto each horse, including the one they just dismounted, and the extras will go home. The merry-go-round will then rotate by one horse so that all the kids on the second horse will get onto other horses as described above. This process will repeat forever.

Eventually, the same configuration of children will repeat. A configuration is considered the same if there are the same number of kids on each horse. All the horses are indistinguishable, so we compare configurations starting from the next horse in each, instead of tracking the original first horse.

From a starting configuration, the configuration of children will eventually repeat. After the configuration repeats, it will stay in a cycle that infinitely repeats the same sequence of configurations. Given the initial configuration of children, find the period of the cycle. Also find the number of configurations, possibly including the initial state, that will never be repeated.

SHORT NAME: merry

INPUT FORMAT:

The first line of input contains N , the number of horse on the merry-go-round a ($1 \leq N \leq 150$).

The second line contains the number of children on each horse, starting from the next horse ($1 \leq a_i \leq 150$), separated by spaces.

OUTPUT FORMAT:

First print the number of steps in the cycle, followed by a space, and then the number of configurations that are never repeated.

Note: You can assume that both answers will be less than 20,000.

SAMPLE INPUT:

```
4
1 5 2 9
```

SAMPLE OUTPUT:

```
2 4
```

The sequence of configurations is:

```
1 5 2 9
0 6 2 9
1 1 3 10
2 2 0 11
3 3 1 1
0 4 2 2
1 1 3 3
```

Note that 3,3,1,1 and 1,1,3,3 are considered the same configuration, starting from the next horse.