



Problem 2 – Bunny Factory

Bunnies are interesting creatures (and cute as hell). If you put two bunnies in a single bordered space and given some time, they will become four, and then eight, and then sixteen, and then thirty two and so on until the whole planet gets filled with little furry balls. Fortunately the humans have put a secret factory to keep this bunny madness under control.

The bunny factory has **N** initial cages filled with bunnies. Each cage has more than 1 bunny (so that they can multiply without a problem). The multiplication process consists of several similar cycles and stop when a certain condition is met. On each **i-th** cycle of the multiplication factory goes through the following steps:

1. If there are less than **i** cages, the factory stops the multiplication process.
2. The factory gets the first **i** cages and calculate the sum **s** of all bunnies in them.
3. If there are less than **s** cages after the **i-th** one, the factory stops the multiplication process.
4. The factory gets the next **s** cages after the **i-th** one and calculates the **sum** and **product** of all bunnies in them.
5. These **sum** and **product** are concatenated as **next** cages. All cages that were not included in the calculations are simply appended to **next**.
6. The factory gets several empty cages then gets the digits of **next** and for each digit puts the same amount of bunnies in each cell. If the digit is **1** or **0**, the digit is ignored.
7. Step 1 is repeated for the newly generated cages with bunnies.

Here is an example for better clarification:

- Consider we have **12** cages with these bunnies: **3 2 5 5 4 8 4 9 5 6 3 4**.
- The factory starts the process with the first cycle, so it takes the first cage as sum **s** – **3**.
- The factory takes the next **3** cages (**2 5 5**) after the first one and calculates their **sum** – **12** – and **product** – **50**.
- The factory concatenates the results and appends all the untouched cages – **12**, **50** and **48495634** – resulting in **125048495634**.
- All digits equal to 1 or 0 are excluded – final result is **2548495634**.
- The factory fills the next cages with bunnies – **2 5 4 8 4 9 5 6 3 4**.
- The factory continues with the second cycle (repeating), so it takes the first two cages and calculates their sum **s** – **2 + 5** – **7**.
- The factory takes the next 7 cages (**4 8 4 9 5 6 3**) after second one and calculates their **sum** – **39** – and **product** – **103680**.
- The factory concatenates the results and appends all the untouched cages – **39**, **103680** and **4** – resulting in **391036804**.
- All digits equal to 1 or 0 are excluded – final result is **393684**.
- The factory fills the next cages with bunnies – **3 9 3 6 8 4**.
- The factory continues with the third cycle (repeating), so it takes the first three cages and calculates their sum – **15**.
- There are no **15** cages after the third one, so the multiplication process stops.
- The final cages are filled with these bunnies – **3 9 3 6 8 4**.

Input

The input data consists of **N + 1** lines – the first **N** lines will be the **N** initial cages filled with bunnies and the last line will always be "**END**" indicating the end of the input.

The input data will always be valid and in the described format. There is no need to check it explicitly.

Output

The output data consists of a single line holding the amount of bunnies in each cage after the multiplication process, separated by a single space character (" ").

Constraints

- **N** will be between **1** and **100**.
- Bunnies in every cage will be between **1** and **100**.
- Some of the calculations during the multiplication process will have **more than 20 digits**.
- Allowed working time for your program: **0.1** seconds.
- Allowed memory: **16 MB**.

Examples

Input	Output
3 2 5 5 4 8 4 9 5 6 3 4 END	3 9 3 6 8 4