Timecard

In IOI 2018, we will record names of person who enters the arena. The record will consist of uppercase letters of the Latin alphabet and/or lowercase ones. To make it easier to read, you will have to convert uppercase letters on the record to lowercase.

Implementation details

You should implement the following procedure and function:

```
init(int N)
```

- This procedure is called only once, before any calls to `convert`.
- `N`: the number of names in the record.

```
string convert(string s)
```

- This function is called N times after `init` is called.
- `s`: the name in the record.
- This function should convert uppercase letters in the string `s` to lowercase, and return the result `t`.

Example

```
init(3)
```

There are 3 names on the record in this example. Then the grader makes the following procedure calls:

Call	Return
`convert("WatanabE")`	"watanabe"
`convert("ITO")`	"ito"
`convert("YamaMoto")`	"yamamoto"

The files `sample-01-in.txt` and `sample-01-out.txt` in the zipped attachment package

correspond to this example. Other sample inputs/outputs are also available in the package.

Constraints

- $1 \leq N \leq 100$
- $1 \leq |s| \leq 20$ (Here $|s|$ is the length of the string s.)

Subtasks

1. (60 points) $N \leq 10$
2. (40 points) No additional constraints

Sample grader

The sample grader reads the input in the following format:

- line 1: N
- line $1 + i$ ($1 \leq i \leq N$): s_i

The sample grader first calls `init(N)`, and then `convert(s)` for $\mathtt{s} = s_i$ ($i = 1, 2, \ldots, N$). It prints the return values of `convert` in the following format (Now, t_i is the return value of `convert(s)` for $\mathtt{s} = s_i$):

- line i ($1 \leq i \leq N$): t_i