You've built an amazing fish-catching robot, but you've discovered one small flaw: the robot can't decide whether to fish or cut bait.

You've designed a language for the Control of Oceangoing Devices (COD) which you plan to use to program the robot. COD has three instructions:

`fish`	**Fish for ten minutes.**
`bait`	**Cut bait for ten minutes.**
`lunch`	**No operation for ten minutes.**

Bait is required to catch fish. The robot must cut bait for twenty minutes (execute two `bait` instructions) to generate a single *bait unit*, enough bait to catch a single fish. Other instructions (`fish` and `lunch`) may be interleaved between the two `bait` instructions that generate a bait unit. The robot has storage for three bait units; it cannot cut any more bait if it is already storing three bait units. If the robot is storing three bait units, a `bait` instruction is treated as if it were a `lunch` instruction (a NOP). Catching a fish consumes a bait unit.

Fish have deterministic behaviour. A fish cannot be caught more often than once every seventy minutes, and after a fish has been caught the robot must fish for thirty minutes before catching another fish (they get shy). In order to successfully complete a `fish` instruction, the robot must have a least one bait unit. If the robot has no bait, a `fish` instruction cannot be successfully completed and is treated as if it were a `lunch` instruction. When the robot first starts fishing, a fish is caught on the first `fish` instruction that completes (beginner's luck). If at least one fish has already been caught then a fish is caught on the completion of a `fish` instruction if and only if: 1) the `fish` instruction is at least the seventh COD instruction executed since the last instruction on which a fish was caught, and 2) the `fish` instruction is at least the third `fish` instruction successfully completed since the last instruction on which a fish was caught.

The execution of a `lunch` instruction allows time to pass, but has no other purpose.

The robot starts with no bait; no fish have been caught.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

Input consists of a sequence of `fish`, `bait` and `lunch` instructions, terminated by the end-of-file.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

Output is a line containing a single integer, indicating the number of fish the robot has caught at the end of the sequence.

Sample Input

```
1

fish
fish
lunch
bait
fish
bait
fish
bait
bait
fish
fish
fish
fish
lunch
lunch
lunch
lunch
fish
fish
fish
```

Sample Output

```
2
```