

TIME ZONES (SWEDEN)

You're a businessman that has customers all over the world. During one day you get *exactly* one message from *each* time zone, including the zone you stay in. The messages come with their sending time specified in them (the messages are transported instantly). Unfortunately due to a millennium bug the senders' local time is given without anything that identifies their time zone.

Task: Your task is to identify from which time zones the messages originated.

In this task the number of hours, n , in one day varies between $5 \dots 60$. The number of time zones is always the same as the number of hours and each time zone has a time displacement which is an integral number of hours.

The zones are numbered from 0 to $n - 1$. You are living and receiving the messages in "GMT", that is in time zone 0 without any time displacement. The time zones are counted westward. That means that you should add z hours to the local time in zone z to get the time in your zone. Note that this is not the common way to count the zones, normally zone 2 would be known as "-2:00".

For example, if the local time in time zone 2 is 03:15 it is 05:15 in your zone ("GMT").

The messages can arrive anytime during the day (that is between 0:00-($n - 1$):59), but no two messages arrive at the same time. The date line goes between zone 0 and the last zone and can thus be ignored.

Input data: The first row of the file ZON.IN contains the number of hours in one day, n , $5 \leq n \leq 60$, which also is the number of time zones and the number of messages you received.

Each of the following n rows contains the local sending time given as hhmm (2 digits for hours and 2 digits for minutes where $0 \leq hh \leq n - 1$ and $0 \leq mm \leq 59$). The rows are ordered in chronological order, i.e. the first message that arrived is on the first row.

There is one unique solution to all given test examples.

Output data: The output should be written to the file ZON.OUT. The file must contain a single row with n numbers from 0 to $n - 1$ identifying in which time zones the n messages originated. The first number corresponds to the first time given in the input etc.

Example: ZON.IN:

```
5
0017
0250
0400
0201
0002
```

ZON.OUT:

```
3 1 0 2 4
```

Note that message 3 must come from time-zone 0 or it would have arrived at a time later than 4:59 which is the last minute in the 5-hour days in this example.