Reminder:

When submitting your solution to the lab system, make sure there is no `package` statement at the top of your `.java` file as this will result in a grade of 0 (restriction of this system).

Hashing (Advanced Problem)

Problem Statement:

Consider a hash table of size N, numbered 0 to N-1. You have to insert integers into this table using the hashing technique given below:

Let *i* be the integer to be inserted. Compute the index *j* of the location where the insertion is to be made as $\boxed{j = i \bmod N}$. If this location is empty then put the element at this position else recompute the next location as follows:

Remove the right most digit of *i* and recompute ***j = i mod N***. If the digit removed was odd, then move *j* locations forward from the current location else move *j* locations backward from the current location (assume 0 as even). Note that this move will wrap around both the edges of the table.

Keep doing this till you either find a free location or all the digits of *i* have been removed. If all digits of *i* have been removed and yet unable to find a free location, then from the last location tried, start moving in the direction corresponding to the last digit removed. Keep moving till you detect a free location.

Assume that the number of integers inserted is not more than the table size.

Input Specification:

The first line will contain just one integer. This will give the table size, N. On the next line will be the list of positive integers that need to be inserted into the table. The integers will be separated by one or more spaces, and the last integer will be -1 indicating end of input. (-1 is not to be inserted into the table).

Output Specification:

The output should contain, for each integer, the locations that were checked while inserting that integer (including the location in which the integer was finally inserted). The locations checked for each of the integers should be output on a line by itself, separated by one space each, each line being terminated by a new line with no spaces before the newline.

Sample Input Output:

Problem 2 (A)

Input: I

```
7
38 52 145 16 179 4 -1
```

7
38 52 145 16 179 4

Output:

```
3
3 5
5 5 4
2
4 0
4 4 3 2 1
```

Note: There are 6 test cases named A to F for this problem. The first one, A, is revealed to you above. You must submit your program for each test case separately, as Problem 2 (A), Problem 2 (B), ..., Problem 2 (F). If your program works correctly for all 6 cases, you have succeeded in solving this programming question.