

# Overrandomized

## Problem

*Note:* Every time this statement says something is randomly chosen, it means "chosen uniformly at random across all valid possibilities, and independently from all other choices".

The company Banana Rocks Inc. just wrote a premium cloud-based random number generation service that is destined to be the new gold standard of randomness.

The original design was that a group of servers would receive a request in the form of a single positive integer  $M$  of up to  $U$  decimal digits and then respond with an integer from the range 1 through  $M$ , inclusive, chosen at random. However, instead of simply having the output written with digits 0 through 9 as usual, the servers were "overrandomized". Each server has a random subset of 10 distinct uppercase English letters to use as digits, and a random mapping from those letters to unique values between 0 and 9.

The formal description of the current situation is as follows: each server has a *digit string*  $D$  composed of exactly 10 different uppercase English letters. The digit string defines the mapping between letters and the base 10 digits:  $D$ 's  $j$ -th character from the left (counting from 0) is the base 10 digit of value  $j$ . For example, if  $D$  were `CODEJAMFUN` then `C` would represent digit 0, `O` would represent digit 1 and `N` would represent digit 9. The number 379009 would be encoded as `EFNCCN` when using that digit string.

When receiving the  $i$ -th query with an integer parameter  $M_i$ , the server:

- chooses an integer  $N_i$  at random from the inclusive range 1 through  $M_i$ ,
- writes it as a base 10 string with no leading zeroes using the  $j$ -th character of  $D$  (counting starting from 0) as the digit of value  $j$ , and
- returns the resulting string as the response  $R_i$ .

We collected some data that we believe we can use to recover the secret digit string  $D$  from each server. We sent  $10^4$  queries to each server. For each query, we chose a value  $M_i$  at random from the range 1 through  $10^U - 1$ , inclusive, and received the response  $R_i$ , a string of up to  $U$  uppercase English letters. We recorded the pairs  $(M_i, R_i)$ . As we were moving these records to a new data storage device, the values of all the integers  $M_i$  within the records of some servers became corrupted and unreadable.

Can you help us find each server's digit string  $D$ ?

## Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow. Each test case contains the records for one server and starts with a line containing a single integer  $U$ , representing that  $10^U - 1$  is the inclusive upper bound for the range in which we chose the integers  $M_i$  to query that server. Then, exactly  $10^4$  lines follow. Each of these lines contains an integer  $Q_i$  (in base 10 using digits 0 through 9, as usual) and a string  $R_i$ , representing the  $i$ -th query and response, respectively. If  $Q_i = -1$ , then the integer  $M_i$  used for the  $i$ -th query is unknown. Otherwise,  $Q_i = M_i$ .

## Output

For each test case, output one line containing `Case #x: y`, where  $x$  is the test case number (starting from 1) and  $y$  is the digit string  $D$  for the server examined in test case  $x$ .

## Limits

Time limit: 20 seconds per test set.

Memory limit: 1GB.

$1 \leq T \leq 10$ .

$D$  is a string of exactly 10 different uppercase English letters, chosen independently and uniformly at random from the set of all such strings.

$M_i$  is chosen independently and uniformly at random from the range 1 through  $10^U - 1$ , inclusive, for all  $i$ .

$N_i$  is chosen independently and uniformly at random from the range 1 through  $M_i$ , inclusive, for all  $i$ .

$R_j$  is the base 10 representation of  $N_i$ , using the  $j$ -th digit from the left of  $D$  (counting starting from 0) as the digit of value  $j$ , for all  $i$ .

### Test Set 1 (Visible Verdict)

$Q_i = M_i$ , for all  $i$ .

$U = 2$ .

### Test Set 2 (Visible Verdict)

$Q_i = M_i$ , for all  $i$ .

$U = 16$ .

### Test Set 3 (Visible Verdict)

$Q_i = -1$ , for all  $i$ .

$U = 16$ .

The sample input is too big to display inline, so we are providing downloadable files instead for the [input](#) and [output](#).