

# Hexacoin Jam

## Problem

The Code Jam team's first cryptocurrency, jamcoins, never caught on. This year, we are trying again with *hexacoins*, which are named for their use of base 16. To "mine" a **D**-digit hexacoin, one has to work with integers using exactly **D** base 16 digits, including leading zeroes if needed.

Each value represents an integer between 0 and  $16^D - 1$ , inclusive. Base 16 digits are represented by the numbers 0 through 9 and the uppercase letters A through F, as usual. For example, F2B, 0C8 and 000 are valid values when **D**=3, corresponding to the base 10 values 3883, 200 and 0. On the other hand, 1234, DF, C0DE and JAM are not valid values when **D**=3.

When performing addition of **D**-digit base 16 values, any overflow digits are dropped. That is, the addition is performed modulo  $16^D$ . For example,  $F2B + 0C8 = FF3$  (4083 in base 10) and  $F2B + F2B = E56$  (3670 in base 10, because the sum's result is 7766, and taking modulo  $16^3$  yields 3670).

To "mine" a **D**-digit hexacoin, a computer must perform the following steps:

1. Choose a list **L** of **N** **D**-digit base 16 values  $L_1, L_2, \dots, L_N$ .
2. Choose a target range of **D**-digit base 16 values: the numbers from **S** to **E**, inclusive.
3. Choose a permutation **P** of the base 16 digits 0 through F, uniformly at random from among all  $16!$  such permutations.
4. Apply **P** to all digits of all numbers in the list, creating a new list **L'** consisting of **N** **D**-digit base 16 values. Formally, the *j*-th digit of the *i*-th element of **L'** is the result of applying **P** to the *j*-th digit of the *i*-th element of **L**.
5. Choose a pair of elements from **L'** without replacement, uniformly at random from among all such possible choices, and independently of the choice of permutation.
6. Calculate the sum (dropping overflow digits) of the two chosen elements.

If the sum calculated in the last step is between **S** and **E**, inclusive, then a hexacoin has been found! For example, suppose that:

- **L** = [134, 000, FFB, 000, AA9].
- **S** = 85C and **E** = EDF.
- The computer happens to choose  $P = (0 \rightarrow 4, 1 \rightarrow A, 2 \rightarrow 2, 3 \rightarrow 8, 4 \rightarrow 9, 5 \rightarrow B, 6 \rightarrow C, 7 \rightarrow 7, 8 \rightarrow F, 9 \rightarrow 1, A \rightarrow 0, B \rightarrow 3, C \rightarrow 5, D \rightarrow 6, E \rightarrow E, F \rightarrow D)$ .

Then, when **P** is applied to **L**, the resulting **L'** is [A89, 444, DD3, 444, 001]. Notice that **P** is not applied to **S** and **E**.

There are  $(5 \times 4) / 2 = 10$  pairs of values to choose, and each pair has a probability of  $1/10$  of being chosen. The only sums that fall within the range are  $A89 + DD3 = 85C$ ,  $444 + 444 = 888$ ,  $A89 + 001 = A8A$ ,  $DD3 + 001 = DD4$ , and  $A89 + 444 = ECD$  (twice).

The first two steps are already computed and you know the list **L** and the range [**S**, **E**] that were chosen. What is the probability that a hexacoin is found after the rest of the process is performed?

## Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of three lines. The first line contains two integers **N** and **D**: the size of the given list and the number of digits to work with, respectively. The second line contains two **D**-digit base 16 numbers **S** and **E**: the inclusive lower and upper bounds of the target range, respectively. Then there is one more line containing **N** **D**-digit base 16 numbers **L<sub>1</sub>**, **L<sub>2</sub>**, ..., **L<sub>N</sub>**, representing the values in the list.

## Output

For each test case, output one line containing Case #*x*: *y* *z*, where *x* is the test case number (starting from 1) and *y* and *z* are non-negative integers, such that the fraction *y*/*z* represents the probability of finding a hexacoin, under the conditions described above. All of *x*, *y*, and *z* must be in base 10. If there are multiple acceptable values for *y* and *z*, choose the ones such that *z* is minimized.

## Limits

Time limit: 90 seconds per test set.

Memory limit: 1GB.

$2 \leq N \leq 450$ .

**S** contains exactly **D** characters.

Each character of **S** is a base 16 digit.

**E** contains exactly **D** characters.

Each character of **E** is a base 16 digit.

$S \leq E$ .

**L<sub>i</sub>** contains exactly **D** characters, for all *i*.

Each character of **L<sub>i</sub>** is a base 16 digit, for all *i*.

### Test Set 1 (Visible Verdict)

$1 \leq T \leq 100$ .

$2 \leq D \leq 3$ .

### Test Set 2 (Hidden Verdict)

$1 \leq T \leq 100$ .

$2 \leq D \leq 4$ .

### Test Set 3 (Hidden Verdict)

$1 \leq T \leq 10$ .

$2 \leq D \leq 5$ .

## Sample

### Sample Input

```
4
2 2
10 10
00 FF
2 2
10 11
```

### Sample Output

```
Case #1: 7 120
Case #2: 1 15
Case #3: 0 1
Case #4: 2731 8736
```

```
00 FF
4 3
FFF FFF
230 A10 010 F70
4 3
AFF FFF
230 A10 010 F70
```

In Sample Case #1, the target range is just a single value 10. Since the result ends with 0, the sum of the values assigned to both last digits 0 and F must end in 0 as well. Since  $P[0]$  and  $P[F]$  are different values, their sum cannot be exactly 0. Therefore,  $P[0] + P[F]$  must be 10 (in base 16). There are 7 pairs of different digits that accomplish that.  $P[0]$  and  $P[F]$  cannot both be 8. All 7 pairs lead to an overall sum of 10 (after dropping an overflow 1). Therefore, there are 14 assignments of different digits to 0 and F that lead to a hexacoin. There are  $16 \times 15$  possible assignments to those digits, so the result is  $14/240 = 7/120$ .

In Sample Case #2, we need to add the probability of the result being exactly 11 to the result of Sample Case #1. The only way that happens is if 0 and F are assigned to 0 and 1, in either order. That has a probability of  $2/240 = 1/120$ , leading to a total of  $7/120 + 1/120 = 8/120 = 1/15$ .

In Sample Case #3, notice that regardless of which permutation and pair of numbers the computer chooses from the list, we will add two numbers that end in the same digit. That produces an even result, even after taking it modulo  $16^3$ . Since the only value in range is odd, we have no hope of mining a hexacoin in this case. Notice that  $0 \ 2$  is an invalid representation of the answer because  $z$  would not be minimum.