

Martia Auris 3020

DNA is a long chain of molecules that contains all the information necessary for the life functions of a cell. The individual molecules that make DNA are called nucleotides. There are only four nucleotides that are ever used. These are Adenine **A**, Thymidine **T**, Guanine **G**, and Cytosine **C**.

As the cells in your body (and all other organisms on Earth) undergo constant divisions and are susceptible to mistakes, the DNA sequences can mutate easily. This can happen because you put your hand too close to the core of a nuclear reactor or just because your cells made mistakes on their own.

In the year 3020, a new deadly superbug, Martia Auris, was discovered in a human habitat on Mars. Superbug is a term for bacteria strains, viruses, or fungi resistant to most antibiotics and other medications commonly used in the 31st century. Martia Auris is a bacteria strain conducted only by humans, is highly infectious, gives symptoms very late, and the mortality rate after infection is high.

The martian laboratories collected samples and analyzed the bacteria genome. The dataset consists of pairs of two sequences of the same length:

1. Top one is the sequence we expect (**E**),
2. The second one is bacterial sequence found in the patient tissues (**P**).

You need to create a simple tool for laboratories to search the collected data and identify mutations in the bacteria.

Sequence format

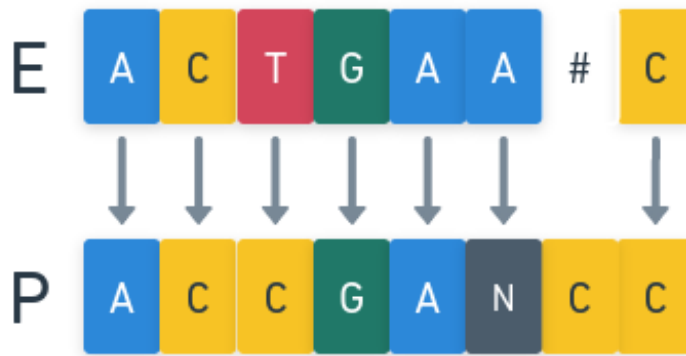
Martia Auris mutates slightly, so the found sequences nearly always differ. The sequences consist of letters **A**, **C**, **T**, **G**, **#**, and **N**, and they're always of the same length. For each position i in the sequence:

- If $E[i] == "N"$ (or $P[i] == "N"$), then we assume there is nucleotide A, C, T, or G on an i -th position in the E sequence (or P accordingly), but we do not know which one due to experiment uncertainty. We assume the best-case scenario, so "N" always matches all other nucleotides. You can think of it as "." in regex or a wildcard that matches all nucleotides.
 - If $E[i] == P[i]$ then no mutation occurred on i -th position
 - If $E[i] == "#"$ && $P[i] != "#"$ then new nucleotide $P[i]$ appeared inside patient sequence
 - If $E[i] != "#"$ && $P[i] == "#"$ then a nucleotide $E[i]$ on i -th position was removed by the new mutation
 - If $E[i] == "#"$ && $P[i] == "#"$ case is not allowed, because it's meaningless
 - In other case nucleotide $E[i]$ was mutated into different nucleotide $P[i]$ by a mutation
-

Example sequences

For example for pair:

E: ACTGAA#C
P: ACCGANCC



We have three changes:

1. **T** was mutated into **C** (3rd position),
2. **A** was potentially mutated into something (6th position),
3. **C** was inserted (7th position).

Now the task is to read all pairs of patient samples readings and for each pair:

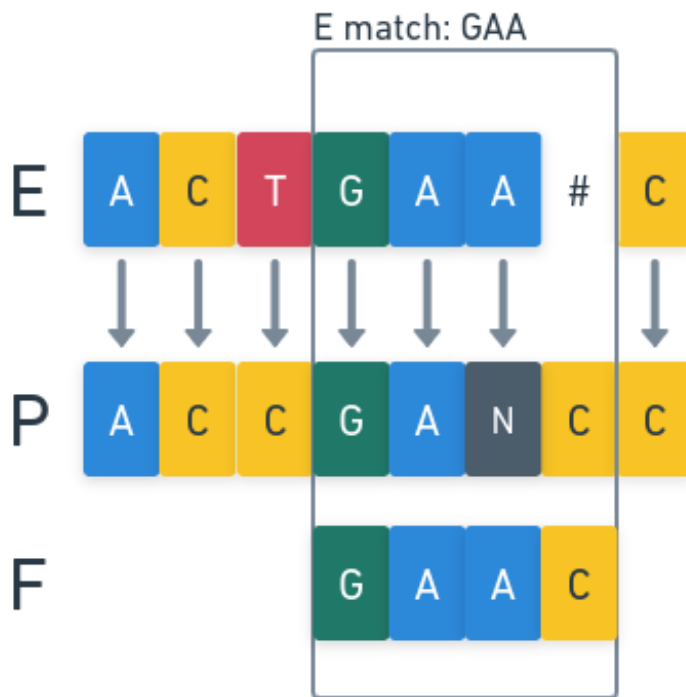
1. Find each occurrence of the input sequence (we call it **F**) in the patient sample sequence **P**. Note that **F** can contain **N**, but not **#**.
2. Print fragment of **E** corresponding to the found fragment (we translate every single match of **F** in **P** to the original fragment in **E**).
3. All output sequences should be lexicographically sorted. Note that in some cases, we can have duplicate sequences in the program output, and that's normal.

Important note: we don't care if **F** is present in **E**. We want only matches in **P** sequences but translated later into corresponding fragments from **E** sequence. We don't care about occurrences of **F** in **E**.

Example 1

Let's look at the example:

F: GAAC
E: ACTGAA#C
P: ACCGANCC



Then the patient sequence contains "GANC". **N** can be anything, so we qualify that as a result. That fragment before mutations would be "GAA" (because **C** was inserted by mutation and **A** was changed into **N**)

The output would be:

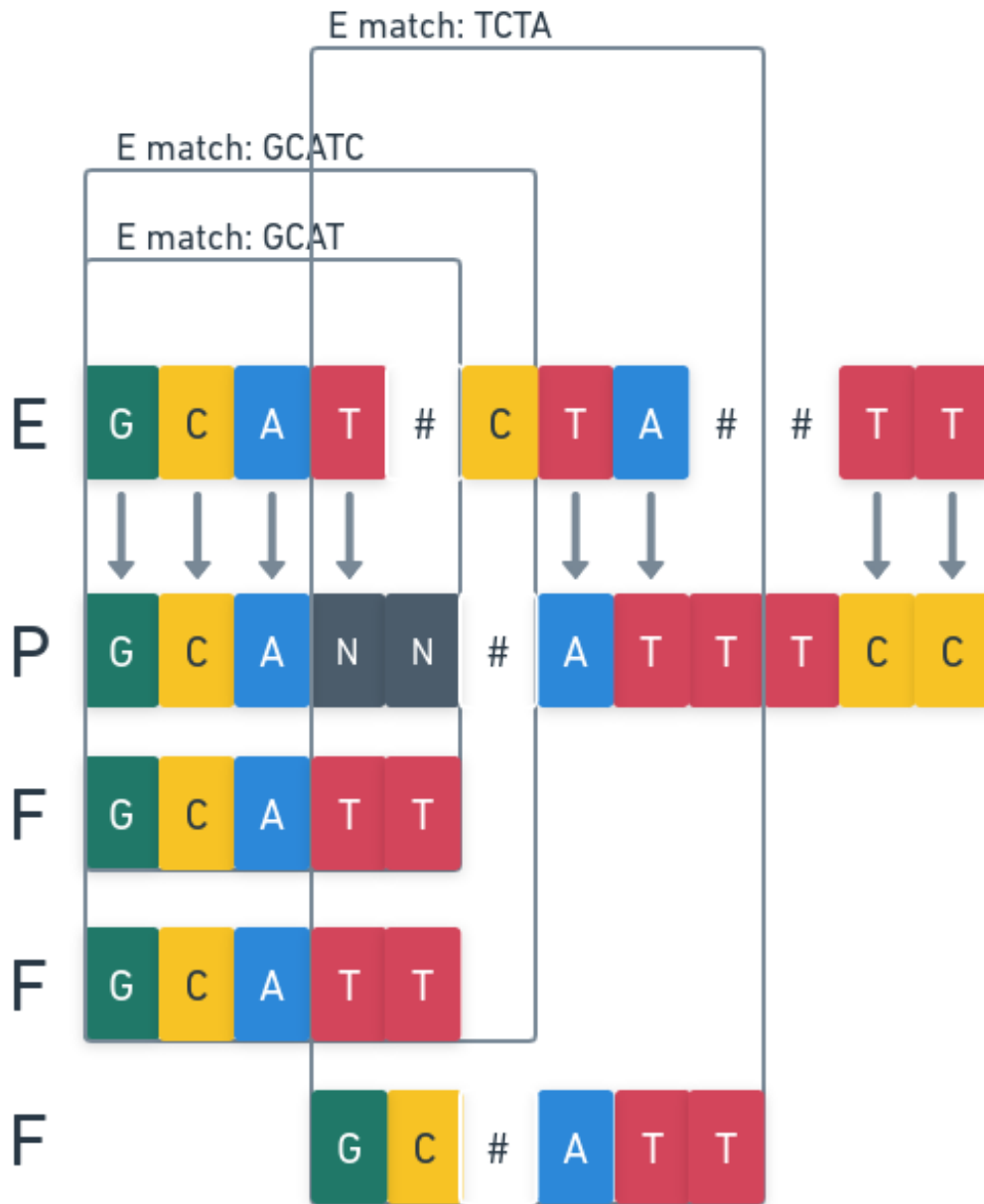
GAA

Note that the **E** sequence contains the **F** sequence ("GAA#C"), but as noted earlier, we don't care about **F** matches in **E**. That's why the output doesn't contain "GAAC". We do only care about **F** matches in **P** translated to the corresponding fragments in **E**

Example 2

Another example:

F: GCATT
E: GCAT#CTA##TT
P: GCANN#ATTTC



The patient sequence contains "GCANN", **N** can be anything, so we qualify that as a result. The fragment before mutations would be "GCAT" (**T** was changed into **N**, and new **N** appeared). The important note here is that we also qualify "GCATC" as a result because the last nucleotide **C** will be removed, and we also end up with the same sequence.

The patient sequence also contains "NN#ATT". **#** represents only that something was removed, so we get "NNATT" which similarly can be qualified as a result. The fragment before mutations would be "T#CTA#" which is "TCTA" (**T** was changed into **N**, new **N** appeared, **C** was changed into **A**, and **A** into **T**).

The output should be:

```
GCAT
GCATC
TCTA
```

Input

The first line of input contains the sequence **F** we are looking for (note that sequence F can

contain **N** but not **#**). The second line of the input contains the number of pairs we should read. The following lines contain all pairs of **E/P** sequences.

Output

The output should contain in each separate line an original sequence in **E** corresponding to sequence **F** found in sequence **P**. The output should be lexicographically sorted and can potentially contain duplicates.

Examples

Input

```
GAAC
2
ACTGAA#C
ACCGANCC
GAACGANC
GAACGANC
```

Output:

```
GAA
GAAC
GANC
```

Input

```
GCATT
1
GCAT#CTA##TT
GCANN#ATTTCC
```

Output:

```
GCAT
GCATC
TCTA
```

Input

```
NN
1
ACA
A#A
```

Output:

```
ACA
```