

Career Fair 2014

Programming Competition

Biologists have sequenced the genes of a pre historic family of organisms and they have stored the sequences in a text file represented by characters. Each line of the file contains the gene sequence of one organism. They also have learned that these family of organisms came from one organism whose gene sequence is stored in the first line of a text file. The biologist have learned that the gene sequence of the organism can change in one of the 4 ways every time it mutates.

1) two of its genes may swap

e.g: abcdelmnop -> abcedlmnop

2) one gene may vanish

eg: abcdelmnop -> abcdlmnop

3) a new gene may get introduced

eg: abcdelmnop -> abcde#lmnop

4) one gene may be replaced by new one

eg: abcdelmnop -> abcdel*nop

Also they have learned that there will be at most two changes that may occur from one generation to another. For example the following two sequences are not in consecutive generations because there are 3 changes between them.

eg: abdcelmnop -> abcdel#mno1p

Your job is to sort the genetic sequence of the organisms starting from the oldest to the newest organism. Note that there could be an organism gene sequences in the input file that may not be related to the family of organism the biologists are studying. You must identify these sequences too.

Input

A text file with sequences of characters representing the genetic make-up of the organisms under study. Each line represents one organism, and the first line is the oldest organism at the root of the mutation. With this instruction you are provided with a text file input.txt which has the sample data.

```
abcdelmnop!fghijkquxystuvw$elmnop*abcjkquxystuvw1ghi
zbcdelmno!pfg2jkquyxr3stuvw$emlnopabcj#kquxystuvw1hi
zbcdelmno!pfg2jkquyxr3stuvw$elmnopabcjkquxystuvw1hi
abcdelmnop!pfg2jkquyxr3stuvw$elmnopabcjkquxystuvw1hi
zcbdelmno!pkgh2jkquyxr3stuvw$emlnopabcj#kquxystuvw1hi
abcdelmnopfghijkquxystuvwdelmnop*abcjkqu5xystuvw1ghi
abcdelmnop!fghijkquxystuvw$elmnop*abcjkquxystuvw1hi
zcbdelmno!pfg2jkquyxr3stuvw$emlnopabcj#kquxystuvw1hi
zbcdelmno!pfg2jkqu6xr3stu8wd$emlnvpabcj#kquyrtuvwg1hi
zbcdelmno!pfg2jkquyxrstuvw$elmnopabcj#kquxystuvw1hi
zbcdelmno!pfg2jkquyxrstuvw$elmnopabcj#kquxystuvw1hi
```

Output

You should have two versions of your program with different levels of output. The output file should be named output.txt and it should be located in the same folder as the program.

Version 1: In this mode the program should write the steps it takes to sort the gene sequences in the input file. This version of the program will be used to test your algorithm. In this mode the program should output:

1) When the program compares two gene sequences, it should write

comparing x and y

2) When the program finds two gene sequences that are one after the other it should output

x after y

3) When the program changes the order of a gene sequence, it should output

move x to y

4) When the program finds a gene sequence outside of the family, it should write

x not in family

Where x and y are the line number of the sequence starting from 1.

Version 2: In this mode, the program will be tested for speed. The program should write the sorted gene sequences from the oldest to the newest generation into a text file name output.txt. Any gene sequence that is not related to the family should be placed at the end of the text file below a marker that says "NOT IN FAMILY". See example below.

```
abcdelmnop!fghijkquxystuvw$elmnop*abcjkquxystuvw1ghi
abcdelmnop!fghijkquxystuvw$elmnop*abcjkquxystuvw1ghi
abcdelmnop!pghijkquxystuvw$elmnopabcjkquxystuvw1ghi
zabcdelmnop!pghijkquxystuvw$elmnopabcjkquxystuvw1ghi
zabcdelmnop!pghijkquxystuvw$elmnopabcj#kquxystuvw1ghi
zabcdelmnop!pgh2jkquxystuvw$elmnopabcj#kquxystuvw1ghi
zabcdelmnop!pgh2jkquxystuvw$emlnopabcj#kquxystuvw1ghi
zcbdelmnop!pgh2jkquxystuvw$emlnopabcj#kquxystuvw1ghi
zcbdelmnop!pgh2jkquxystuvw$emlnopabcj#kquxystuvw1ghi
NOT IN FAMILY
zabcdelmnop!pgh2jkquxystuvw$emlnvpabcj#kquxystuvw1ghi
abcdelmnop!pghijkquxystuvw$elmnop*abcjkquxystuvw1ghi
```

Rules:

- You have to use Java SE language
- You have to submit all your source code files and compiled files for the two output versions
- Start Date: February 27, 2014
- End Date: March 9, 2014 at 9 pm

