

3106 Longest Subsequence

In genetics, contemporary techniques for sequencing an organism's entire genome frequently involve 'cutting up' an entire chromosome into DNA sequence fragments.

Each fragment is then sequenced into the four letters of the genetic code — the DNA bases **A**, **C**, **G**, **T** — that are arranged into genes and non-coding sections.

Afterwards, the multiple fragments are reassembled into one continuous stretch representing a single chromosome.

Your challenge is to find and implement an algorithm for reassembling multiple fragments of DNA strands (strings) into the longest possible sequence.

Two strings, s_1 and s_2 are said to **CONNECT** or **overlap** if the last x characters from the *end* of s_1 are the same as the first x characters of s_2 .

The strings $s_1 = \text{'ACT'}$ and $s_2 = \text{'CTG'}$ overlap because the last two characters of s_1 , 'CT' match the first two characters of s_2 .

Two strands *always* overlap or connect up to the 'deepest' possible point. For example, the string "ACTCT" and "CTCTG" are said to overlap as "A(CTCT)G" and *not* simply as "ACT(CT)CTG".

Other examples:

s_1	s_2	Overlap?
AC	AC	(AC)
AC	CTG	A(C)TG
ACT	CTG	A(CT)G
ACTG	CTG	A(CTG)
ACT	GCA	- no -
ACCC	CCTG	AC(CC)TG
ACTCT	CTCTG	A(CTCT)G

Given a set of *unique* input strings $S = \{s_1 \dots s_n\}$, for example:

```
ACTC
ATG
CGT
TCAG
GTCG
```

The program should find the longest possible sequence, L composed of any m connected strings from the set S such that L can be constructed by concatenating the relevant strings, in proper order and combining the overlapping substrings. Note that you can only use each input string once, and only once to construct L .

Again, note that the problem excludes the possibility of duplicated input strings.

For example, given the above set of strings, L would thus be "ACTCGTCG" or:

```
ACTC
  CGT
    GTCG
```

Note: For any given set S of strings, there may be *more than one* 'longest' sequence possible.

Input

The first line in the input file will be the number of input sets to solve for.

The next line is a number $2 < x \leq 10$ specifying the number of strings in the first input set. The following x lines will be arbitrary strings where the number of characters in each string l will be $3 < l \leq 10$. Each string may consist of any possible combination of l characters 'A', 'C', 'T', 'G'.

Another number, and the corresponding number of strings may follow which will constitute the second set, and so on until the end of the file. There will be at least one (1), but at most four (4) input sets in the input file.

Output

For each set i defined in the input, the program should output one line stating 'Found n longest sequence/s for set i :' followed by n lines where each corresponds to one of the n longest sequence/s found.

Note: If there is more than one possible sequence with the longest length that can be constructed as such given the input set, then the program should output *all* possible 'longest' sequences *in alphabetical order*.

Sample Input

```
2
5
ACTC
ATG
CGT
TCAG
GTCG
9
AGAA
CTTT
GGA
TAGCTG
AAGGGGG
TCCAC
CCAGTCTG
TACG
TATC
```

Sample Output

```
Found 1 longest sequence(s) for set 1:
ACTCGTCAGTCG
Found 2 longest sequence(s) for set 2:
TATCTTTCCACCAGTCTGGAGAAGGGGG
TCCACTTTATCCAGTCTGGAGAAGGGGG
```