

Problem C: DNA Sequences

Introduction

“We wish to suggest a structure for the salt of deoxyribose nucleic acid (D.N.A.). This structure has novel features which are of considerable biological interest.”
J. D. Watson & F. H. C. Crick, *Nature*, April 25th, 1953

April 25th this year is the 50th anniversary of the seminal article of Watson and Crick in *Nature* that described the structure of DNA, the molecule that contains the genetic information of every living being on Earth. Their discovery was a breakthrough in biology, and it made possible numerous advances in understanding living organisms, including humans. Fifty years later, the Human Genome Project gave us a map of our genes, possibly opening the way for better drugs and treatments.

The success of the Human Genome Project was in large part due to the widespread use of computers and sophisticated software. Mapping the DNA requires the efficient manipulation of massive amounts of data. Computational biology and bioinformatics are new areas of computer science that deal with this kind of problem.

The genetic information in a DNA molecule is coded as a sequence of *bases*. There are four different bases: adenine (A), cytosine (C), guanine (G), and thymine (T). Determining the sequence of bases in a given piece of DNA is called *sequencing* the DNA. Your job is to write a program that helps us in this task.

We have a long piece of DNA that we have to sequence. We have made several measurements, each measurement describes the sequence of bases for some small segment of the DNA. Each such segment has the same length ℓ . The first segment contains the first ℓ bases of the sequence. The second segment starts at the $(\ell - 4)$ th base, it overlaps with the first segment on 5 bases. The third segment overlaps with the second segment on 5 bases, and so on. For example, if we have this DNA sequence of 49 bases:

ATTCGTACCGGAGTCCCAGACCTCGGGTTAAACACATATAGATGCAGAT

and $\ell = 16$, then we have the following 4 measurements:

ATTCGTACCGGAGTCC	TCGGGTTAAACACATA
AGTCCCAGACCTCGGG	ACATATAGATGCAGAT

Unfortunately, due to some software errors, the order of the segments became mixed up. So we have all the segments, but we do not know which segment corresponds to which measurement. Given the segments obtained by the measurements (in some random order), your program has to recover the original sequence.

Input

The input is a text file. The first line contains two integers n and ℓ , where n is the number of segments in this test case and ℓ is the length of the segments. The next n lines describe the n segments obtained by the measurements (in random order). Each line contains a string of length ℓ , each letter of the string is one of 'A', 'C', 'G', or 'T'.

Output

You have to output the original sequence in a file whose length equals the length of the original sequence. (Notice that a simple calculation shows that the length of the original sequence is $n(\ell - 5) + 5$). The output file should contain only the 'A', 'C', 'G', or 'T' characters. Do *not* terminate the file with a new line character. If there are multiple possible solutions (the original sequence cannot be unambiguously recovered), then you can output any correct solution.

Sample Input

sample-C-1.in:

7 12

CACAGTGAGGCT

AGGCTTCAAGCA

TTAGAACCATCC

GGAGGCCCACAG

CATCCTTAGGCT

AGGCTATGGAGG

AGGCTTATTAGA

Sample Output

There are two correct possibilities for
sample-C-1.out:

AGGCTATGGAGGCCCACAGTGAGGCTTATTAGAACCATCCTTAGGCTTCAAGCA

and

AGGCTTATTAGAACCATCCTTAGGCTATGGAGGCCCACAGTGAGGCTTCAAGCA