Problem B

DNA subsequences[2]

FASTA archive is a text-based format to store DNA/RNA in witch the bases are represented using single-letter codes. In bioinformatics, this file is used to sequence alignment and string matching.

Write a parallel program to find DNA subsequences (a.k.a. query string) in a FASTA database. If a query string matches within multiple sequences, each result must be reported. If a query string matches multiple locations in the same sequence, the earliest position that matches exactly must be reported.

Input

The input must be read from two different files. Both of them follow FASTA format.

The FASTA format represents many sequences. Each sequence contains one line with the DNA description followed by several lines with the bases. The description line begin with a greater-than character ('>'). The bases sequence are made up of only four characters ('A', 'T', 'C', 'G') and divided by line within 80 characters per line. The file ends with the EOF-mark. The base length is up to 1,000,000 bases.

The database must be read from a file named dna.in

The query file must be read from a file named query.in

Output

The output contains the string matching results. For each query string, the program must output its description in one line followed by its report. If the query string was found within the database, the report contains the sequences description followed by the position within the sequences that exactly match. If the query string is not found within any sequences, a 'NOT FOUND' message must be printed.

The output must be written to a file named dna.out

[2] Based on String Matchin problem from 2009 Intel Threading Challenge.

Example

FASTA database

```
>Escherichia coli partial genome (1)
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTG
TCTGATAGCAGCTTCTGAACTG
GTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCA
>Escherichia coli partial genome (2)
CTAAATACTTTAACCAATATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACA
CAACATCCATGAAACGCATTAG
CACCACCATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTA
CAGGAAACACAGAAAAAAGCCC
GCACCTGACAGTGCGGGCTTTTTTTTTCGACCAAAGGTAACGAGGTAACAACCATGCG
AGTGTTGAAGTTCGGCGGTACA
```

Query file

```
>Query string #1
TATAGG
>Query string #2
TTTT
>Query string #3
ATCG
>Query string #4
AACTGG
```

Output

```
>Query string #1
>Escherichia coli partial genome (2)
17
>Query string #2
>Escherichia coli partial genome (1)
3
>Escherichia coli partial genome (2)
178
>Query string #3
NOT FOUND
>Query string #4
>Escherichia coli partial genome (1)
75
```