Indian Statistical Institute

DFS LAB – COURSE PROJECT 1

MTech(CS) I year 2018–2019

Deadline: 30 November, 2018

(For BONUS marks)

SUBMISSION INSTRUCTIONS

1. Naming convention for your programs: `cs18xx-project1-progy.c` ('y' stands for the subproblem)
2. To submit a file (say `cs18xx-project1-progy.c`), go to the directory containing the file and run the following command from your account on the server (IP address: 192.168.64.35)

   ```
   cp -p cs18xx-assign1-progy.c ~dfslab/2018/project1/cs18xx/
   ```

   If you want to submit your files from a computer with a different IP address, run

   ```
   scp -p cs18xx-project1-progy.c \
   mtc18xx@www.isical.ac.in:/user1/perm/pdslab/2018/project1/cs18xx/
   ```

   (enter your password when prompted).

   To submit all `.c` and `.h` files at one go, use `cp -p *.c *.h ~dfslab/2018/project1/cs18xx/` or similar.

NOTE: All programs should take the required inputs from stdin, and print the desired outputs to stdout.

Q1. **EERTREE:** Let us recall that a palindrome is any arbitrary string $s_1 s_2 \cdots s_n$ equal to its reversal $s_n s_{n-1} \cdots s_1$. The EERTREE, also termed as palindromic tree, is a newly developed and efficient data structure for processing palindromes in strings [1]. It inherits some ideas from the construction of both the suffix trie and suffix tree. This linear-size data structure provides a fast access to all palindromic substrings of a string or a set of strings.

It has been reported that joint EERTREE for several strings can be easily constructed [1]. There are several computational problems (that require the comparison of multiple strings) for which it may be useful to build a joint data structure. For example, a variety of problems can be solved by joint ("generalized") suffix trees (see [2]). Following this, one can build joint EERTREE of a set of strings for solving problems. Some of the problems that can be easily solved by the construction of a joint EERTREE are listed below [1].

(a) Finding the number of subpalindromes, common to all k given strings.

(b) Finding the longest subpalindrome contained in all k given strings.

(c) For the strings S and T, finding the number of palindromes P having more occurrences in S than in T.

(d) For the strings S and T, finding the number of equal palindromes, i.e., of triples (i, j, k) such that $S[i..i + k] = T[j..j + k]$ is a palindrome.

The solution sketches of the four aforementioned problems are highlighted in [1] (see pp. 255). Write separate programs to solve all these problems efficiently.

Palindromic sequences play an important role in molecular biology. They are often found in biological sequences (e.g., DNA, RNA, protein, etc.) and hold important encoded information. Our goal is to find common palindromic sequences and their occurrences in either DNA (nucleotide) sequences or protein (peptide) sequences. Sequences are expected to be represented in the standard FASTA format, with the following exceptions.

- Lower-case letters are accepted and are mapped into upper-case.
- Any numerical digits in the sequence should either be removed or replaced by appropriate letter codes (e.g., N for unknown nucleic acid residue or X for unknown amino acid residue).
- A single hyphen or dash can be used to represent a gap of indeterminate length.

Input Format
The input strings are provided in separate files (names to be read from command line arguments). The number of files are specific to the subproblems. However, for the third subproblem a palindromic string is to be provided. This is to be read from the stdin.

The input files are given in the FASTA format. The FASTA format is a text-based format for representing either nucleotide sequences or peptide sequences, in which base pairs (4 in total) or amino acids (20 in total) are represented using single-letter codes. A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line begins with a greater-than ('>') symbol that distinguishes it from the sequence data. As a standard rule, all lines of sequence data must be of same length (limited to 80 characters in length by recommendation). Some example sequences in FASTA format are show below.

A DNA sequence in FASTA format:

```
>unknown gene|01-01-2018
AACCTGCGGAAGGATCATTACCGAGTGCGGGTCCTTTGGGCCCAACCTCCCATCCGTGTCTATTGTACCC
TGTTGCTTCGGCGGGCCCGCCGCTTGTCGGCCGCCGGGGGGGCGCCTCTGCCCCCCGGGCCCGTGCCCGC
CGGAGACCCCAACACGAACACTGTCTGAAAGCGTGCAGTCTGAGTTGATTGAATGCAATCAGTTAAAACT
TTCAACAATGGATCTCTTGGTTCCGGC
```

A protein sequence in FASTA format:

```
>gi|186681228|ref|YP_001864424.1| phycoerythrobilin:ferredoxin oxidoreductase
MNSERSDVTLYQPFLDYAIAYMRSRLDLEPYPIPTGFESNSAVVGKGKNQEEVVTTSYAFQTAKLRQIRA
AHVQGGNSLQVLNFVIFPHLNYDLPFFGADLVTLPGGHLIALDMQPLFRDDSAYQAKYTEPILPIFHAHQ
QHLSWGGDFPEEAQPFFSPAFLWTRPQETAVVETQVFAAFKDYLKAYLDFVEQAEAVTDSQNLVAIKQAQ
```

Output Format
The outputs (to be printed to stdout) are specific to the subproblems.

References

[1] Mikhail Rubinchik and Arseny M. Shur. EERTREE: An efficient data structure for processing palindromes in strings. *European Journal of Combinatorics*, 68:249–265, 2018. (DOI: 10.1016/j.ejc.2017.07.021)

[2] Dan Gusfield. Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. *Cambridge University Press*, First Edition, 1997.