Msc Informatics IT-22 Data Structures Practical Exam

Divide your roll no by 5 and find remainder. Attempt Question no with remainder

0) Consider two text files. The first one is having a list of employees and their department numbers stored (that is empid, ename, deptid, and there is one record per line). The second one is having the departments along with their names and location (that is deptid, deptname, dlocation). Read the files and construct data structures having all this data (that is empid, ename, deptid, dname, dlocation). Display the list of structures and save this information in another file. Enter the file names as command line arguments

./pgm0 in1.txt in2.txt out.txt

1) Write a C program to achieve the following
   a. Read the name of one input file and one output file through command line.
   b. Read all the words from the input file one by one and store in appropriate data structure making sure there are no duplicates
   c. Traverse the data structure and store the output in the output file.

   ./pgm1 in1.txt out.txt

2) Write a program to do the following
   a) Take an input file name and an output file name from command line where the input file has some words
   b) Read these words one by one from the input file. (string tokenisation not required)
   c) Store these words in a double linked list along with their string lengths
   d) Now traverse the double linked list in reverse direction and delete all those nodes having even string length
   e) Traverse the double linked list after deletion and display all words along with the string length from that.
   f) Also write these words along with the string length to the output file
   g) Free the data structure used.

Example :

Input file
Hello, how are you
I am fine
What are you doing?

Double linked list before deletion (when traversal done from reverse direction)
doing? 6 <-> you 3 <-> are 3 <-> what 4 <-> fine 4 <-> am 2 <-> I 1 <-> you 3 <-> are 3 <-> how 3 <-> hello, 6

Double linked list after deletion (when traversal done from reverse direction)
you 3 <-> are 3 <-> I 1 <-> you 3 <-> are 3 <-> how 3

10

20

30

40

50

List

10 -> 20 -> 30 -> 40 -> 50

Search : 30

Final Output: 30 -> 40 -> 50 -> 10 -> 20