Lab 4: Linked lists

CPSC 1021 – Spring 2020

Introduction

During this lab you will:

1. Practice reading data from text files.
2. Insert data read from an input file into a linked list.
3. Create and submit a tarball.

Part I – Reading from an input file

In this part you will practice reading integers from a text file. To read from a text file, you have to declare a pointer to the FILE:

```
FILE * fptr;
```

Include an <stdio.h> library for file I/O, and <stdlib.h> library, so you can use *exit().* You will then use the following command to open the file for reading, where "*input.txt*" is the name of the input file, and "r" is read mode. The file should be located in the same directory as your program, otherwise you will have to type its full path.

```
fprt = fopen(“input.txt”, “r”);
```

If the file is not found (does not exist), the return value will be NULL, so we usually check to ensure the file you are trying to open exists:

```
if ((fprt = fopen(“input.txt”, “r”) == NULL) {
     // print some error message
     exit(1);
}
```

If the file is found, you will read from the file until you reach the EOF (end of file):

```
 // while we still have items to read
while(fscanf(fptr,"%d", &num) != EOF){
    printf("n is:  %d\n", num);
}
```

One of the files provided to you with this lab's tarball contains C code to read from a file, as well as a file with integers. Compile and run this code to figure out how it works.

Part II. Linked Lists

In this part you will read integers from the file *input.txt* and insert them into a linked list. To do so, you need to complete the existing code by implementing several C functions. It is recommended that you divide your task into several subtasks and complete each step before moving on to the next one.

1. Examine the provided C code (it does not compile because some key components need to be added). Add the missing declarations, etc. and write the functions that were not provided.

2. Complete the *main.cpp* file and thoroughly test your linked list functions by inserting some test data into it. Try to insert at the front, at the end, and delete from the front and from the end.

3. Now add the code to read integers from the file "*input.txt*". Every time you read an integer, create a node by allocating memory, store that integer into the data member of the node, and insert it into the linked list in ascending order (from the smallest to the largest). You will need to write one more function that does that. In that function you will need to traverse the linked list starting from the beginning, looking for an appropriate spot to insert the node. Please notice that integers in the input file will not be sorted.

   ```
   void insertStrategic (const NodePtr);
   ```

When you are done, please submit your linkedlist.cpp, linkedlist.h, and main.cpp files as a zipped tarball.

NOTE: If you submit the archive that cannot be open, or is corrupt, you will not be given the opportunity to redo the work and resubmit. You have one shot to get it right.

What/How to submit

Submit your zipped tarball to canvas to be graded. You do not need to submit the input.txt file. Graders will provide their own input file to test your program.