Lab 3: List-O-Matic
Due: February 23rd 11:59pm

Assignment

In this assignment you will implement **`listomatic`** which reads a set of integers as input and then prints the entire set of integers in 3 different orderings. First it prints the entire list sorted in ascending order. Second it prints the entire list sorted in descending order. Last it prints the entire list in the reverse input order (LIFO - last-in-first-out) There is a sample executable at:

```
/home/users/mharmon/cs277/bin/sampleomatic
```

Run the executable and input an integer and then press enter. Continue to input more integers and type either a non-numeric character or CTL+D when you are finished entering integers.

Requirements for Grading

1. listomatic compiles and runs (20 Points)
2. listomatic continues to read integers from stdin until either CTL+D is pressed or a non-numeric character is typed. (5 Points)
3. listomatic can read any arbitrary number of integers as input. (25 Points)
4. listomatic prints "Read x integers" after input is complete. Output must match the sample executable EXACTLY. (5 Points).
5. list-o-matic prints the entire list of integers *one-integer-per-line* sorted in ascending order by value. Output must match the sample executable EXACTLY (15 Points)
6. listomatic prints the entire list of integers *one-integer-per-line* sorted in descending order by value. Output must match the sample executable EXACTLY (15 Points)
7. listomatic prints the entire list of integers *one-integer-per-line* in reverse order of input. Output must match the sample executable EXACTLY . (15 Points)

Notes & Helpful hints.

A *makefile* has been provided for easy compilation. To compile your executable type

```
>make
```

if there are no errors, this will produce an executable called `listomatic`. To run the executable type:

```
>./listomatic
```

To delete binary files and cleanup the directory type:

```
>make clean
```

For your implementation, a double-linked list is a straightforward solution. Your starting point for that implementation might be defining a struct for a node in the double linked list:

```
typedef struct dll_node {
  int value;
  struct dll_node *prev;
  struct dll_node *next;
} dll_node_t;
```