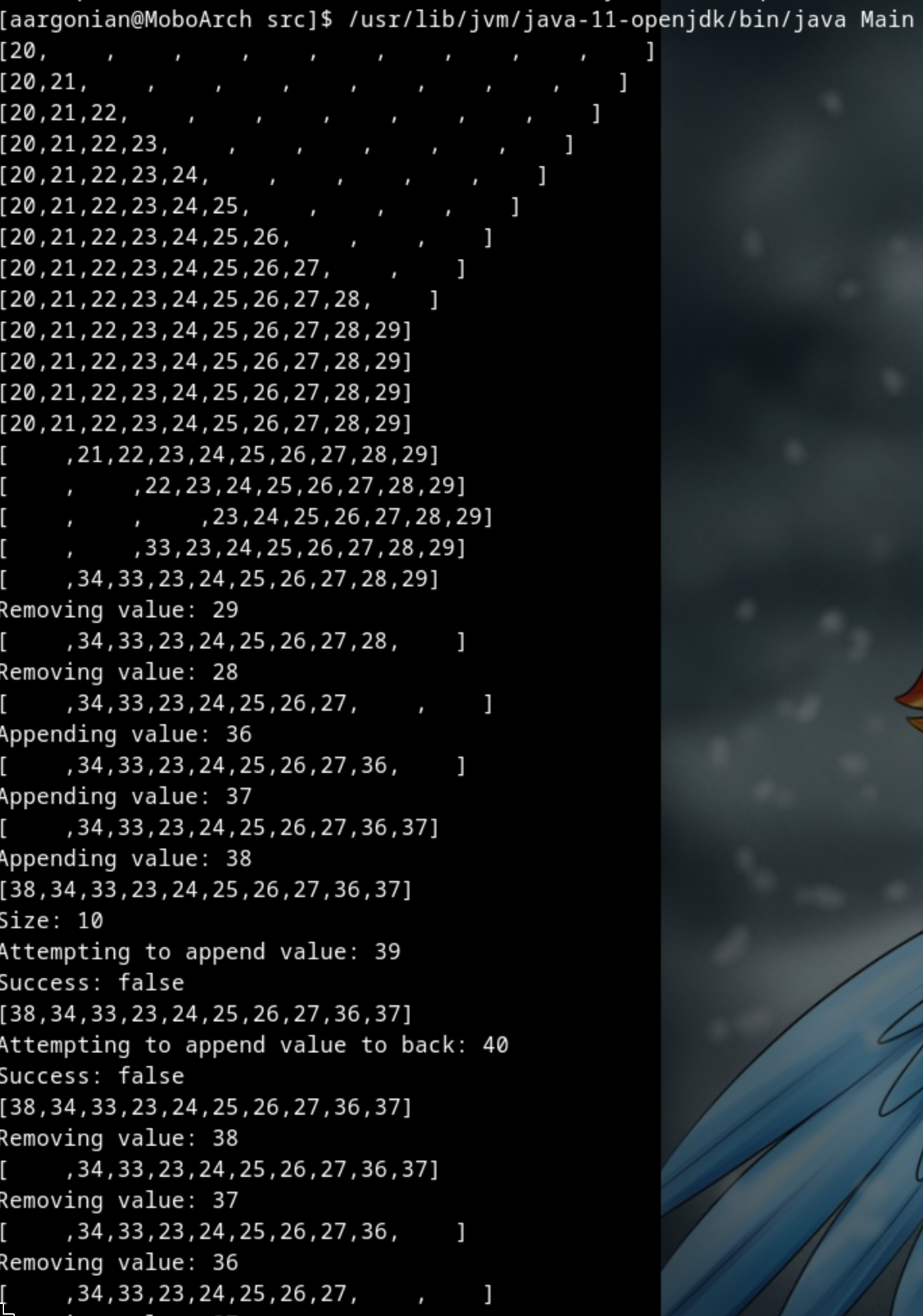
REPORT

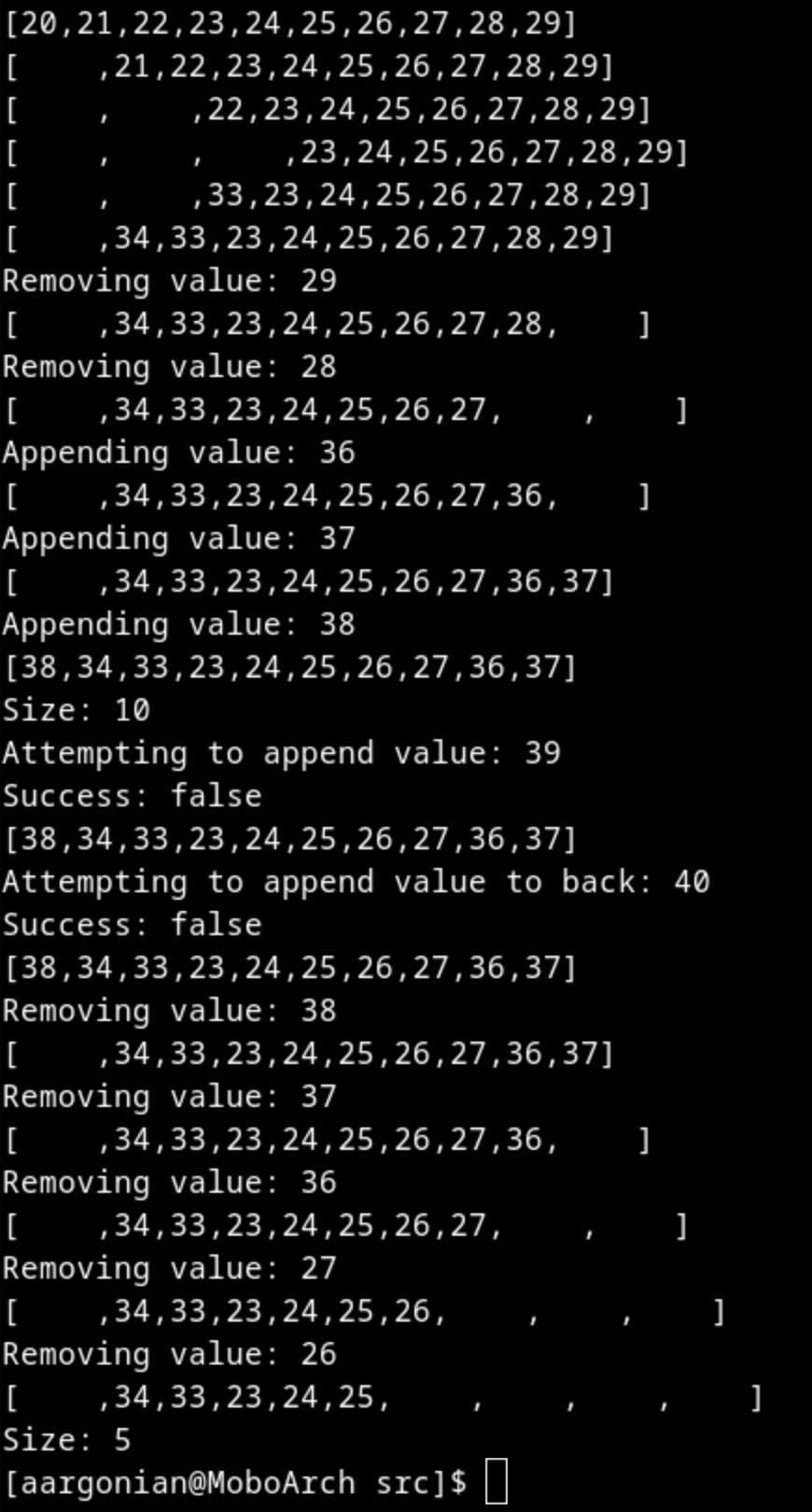
CircularQueue:

This program was implemented to demonstrate the implementation of a circular, double ended queue (deque), utilizing all of its functions.

Although this data structure is termed a queue, it has functions for adding and removing elements at both ends of the structure, effectively removing it’s ability to fully be a “queue”.

The following two images shows a demonstration of the program running. (Note that the images had to be shrunk to fit in the page.



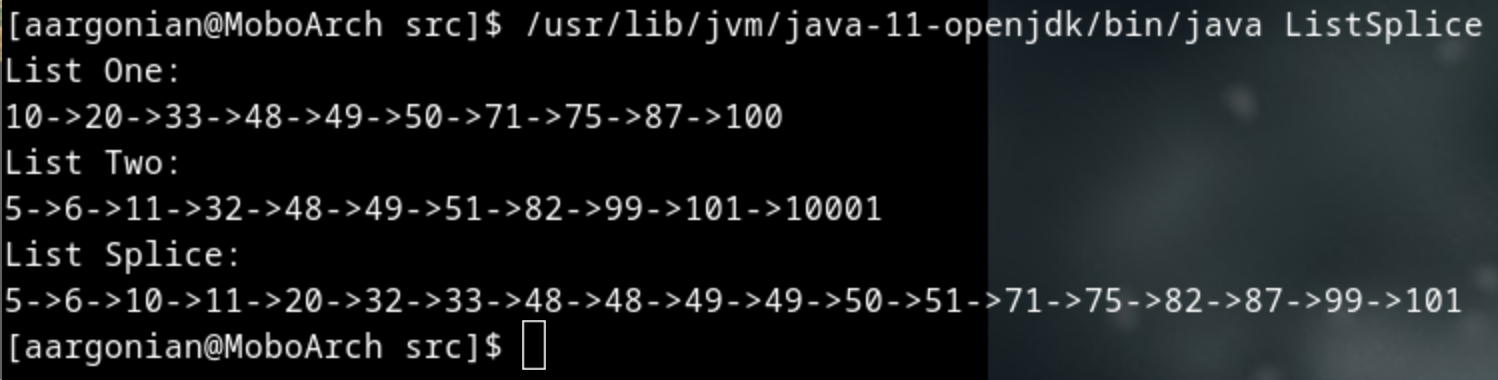


ListSplice:

The next program I implemented had the capability of taking two fully sorted, singly-linked lists and merging them together into one still-sorted singly-linked list.

There isn’t a lot to say about this one. The implementation is pretty standard, simply performs comparisons until one of the lists becomes empty. It then dumps the remainder of the other list into the result and returns it.

Below is a screenshot of a demonstration.



QueueList:

Finally, I implemented a queue as a singly-linked list. The list implementation I used allows for enqueueing, dequeuing, and searching the queue for any object. This one was implemented using Java Generics, and so is capable of storing any object type. Further, it can also reports its length, whether it is empty or not, and return simply the first element without dequeueing it (Similar to peek()).

Below is a screenshot of the program run:

