John Abrams
Chris Zachariah
Asst1: ++malloc
3/10/2019

Design:

We went with a space efficient design in hopes to retrieve partial credit. In the end, we were able to manage a metadata size of just 8 bytes. Each metadata stored 4 short integers: an in–use variable that also serves as a magic number of sorts, the size of its corresponding request (in bytes), and the indices of the meta data previous and next to it in the list. Additionally, the very first byte of our array is a magic letter that informs the program as to whether or not this is the first time that malloc has been called. While we also utilize structs, they are mostly auxiliary and are never directly stored within our array. They can be thought of as “standing alongside” the array.

Workload:

Unfortunately we were not able to completely resolve all of the issues brought up in the specification and included in the workload. Nevertheless, we were able to pass may of the test cases using our highly space–efficient design.