

Project : Heap Manager

1. Extensively testing the (memalloc) and (memfree) operations

- Allocate 1 block -> free -> reallocate, check for the return address:
 - i. Allocation begins from the beginning of the managing list -> the same address after allocation.
 - ii. Allocation begins after the last allocated block -> the address should be altered.
- Allocate multiple blocks -> free 1 block -> reallocate, check for the return address as the previous point.
- Allocate the whole heap size:
 - i. Try to allocate again, it should return NULL address (or may be assertion).
 - ii. Free one segment; try to allocate another with bigger size -> NULL return address.
 - iii. Free two adjacent segments; try to allocate a segment with size bigger than any but less than sum of the two -> allocation succeeds.
- Try to allocate with 0 and negative sizes -> return NULL address.
- Try to free within-segment address -> nothing or assertion.
- Try to free NULL pointer/non-allocated address -> nothing.

2. Bonus:

- Testing the aggregate function: Full regression test with random size allocations with certain rate per second, allocation are held for random period, and then freed. Overall heap should be utilized at one point. Determine the performance of the heap in terms of how many allocations per second can be performed acceptance probability of request as function of request rate per second and average allocation request size.
- Thread-safe operation (develop an example where many threads are created and demonstrate that these threads are working in parallel without errors)/(check critical section determination).