Task 2 report:

To modify the implementation so it follows best fit instead of first fit, I had to change the way find_free_block functions. I keep track of two pointer blocks, current and best. The new implementation loops through the linked list and checks two conditions. The outer condition (current->free && current->size >= size) checks whether the current block is free and that its big enough to fit the request size. If these are true, the current block is a candidate to be the “best”. The inner condition (!best || current->size < best->size) checks to see if the candidate block is smaller than the block that’s currently the “best” or if there is no “best” yet (i.e. the first block in the list). If these conditions are true, the current block overwrites the best block since best fit finds the smallest block that can store the requested size. Finally, the current block is incremented to the next node in the list.