Programming Assignment 3

Due by 2018년 10월 13일 오후 9시

1. Implement `new_node()`, `add()`, `delete()` and `print_list()`, and submit "backend3.c". DO NOT CHANGE ANYTHING ELSE!

2. For this programming assignment, we implement the address book with a linked list. The records are stored in the linked list in an alphabetical order. As always, `data` is the pointer to the linked list that stores the data for the address book.

3. We implement our own memory management as discussed in the class: Allocate a chunk of memory using an array `pool[MAX_RECORD]` and initialize it with `init_pool()` to make it another linked list. And `new_node()` and `free_node()` do the memory management. Note that `MAX_RECORD` is defined to be 5. So, make sure to check if overflow occurs with your implementation of `new_node()`. If an oveflow occurs, your program quits with `exit(0)`.

4. Note that we don't have `search_index()` this time. Why??? Functions `add()`, `search()`, `delete()` and `print_list()` are supposed to examine the nodes one by one from the first node until the task is accomplished.

5. Starting from this assignment, we directly deal with pointers. Be very careful.