Algorithm for MY Favorite Animals Program(Prog 5)
Siqi Wen CS162

1. Create a class (class animal) with some member functions and five data members for the animal name(a pointer); the animal breed(a pointer); what do you like about the animal(a pointer); number of likes(an integer) and nickname you'd give to the animal(a pointer).

2. Create a struct that includes two data members: an object of the class and a pointer of its own type.

3. In the class, there is a constructor function that will initialize all data members and a destructor function that will deallocate the dynamic memory used by the class.

4. In the class, there is a function(called "get_info") that takes a pointer of the struct type as an argument (the pointer will be passed by reference). In this function, create a new node with this pointer(passed as an argument), create a temporary array of size 100 for user's input. Prompt the user for the information of one animal and store the input in the temporary array, then based on the array length, set the dynamically allocated array to the right size, and copy the information stored in the temporary array to the dynamically allocated array.

5. In the class, there is a function(called “insert”) that takes a pointer of the struct type as an argument (the pointer will be passed by reference). In this function, create a pointer of the struct type, pass this pointer as an argument in the “get_info” function. Then insert the new node based on what the linear linked list is like: If it’s an empty list, then let the head pointer point to this new node and set the next pointer in this node to NULL; else if the specific data member in the new node is bigger than the one in the first node, insert the new node at the beginning of this list (in front of the first node); else (if it’s not either of those two situations), use two temporary pointers, one points to the first node, another points to the second node, and find the specific node that the new node needs to insert in front of.

6. In the class, there is a function(called “add”) that takes a pointer of the struct type as an argument (the pointer will be passed by reference). In this function, ask the user if they want to add a new animal, based on the user’s input, decide whether to call the “insert” function or not.

7. In the class, there is a function(called “display_one”) that takes a pointer of the struct type as an argument (the pointer will be passed by value). This function will display the contents of each animal in the list.

8. In the class, there is a function(called “display_all”) that takes a pointer of the struct type as an argument (the pointer will be passed by value). In this function, display particular information based on what the list is like: if the list is empty, then output the message saying it’s an empty list; if it’s not empty, then display all contents of all animals by calling the “display_one” function in a while loop, and the condition in the while loop is that the item the temporary pointer is pointing to is not NULL.

9. In the class, there is a function(called “display_specific”) that takes a pointer of the struct type as an argument (the pointer will be passed by value). In this function, prompt the user for a breed, then find the animals in the list that match with this breed, then display them.

10. In the class, there is a function(called “remove_by_name”) that takes a pointer of the struct type as an argument (the pointer will be passed by reference). In the function, ask what animal the user wants to remove, and tell them they have to enter an animal name (has to be the same as one of the animal names they entered earlier), get the user’s input, and find if there’s in a match in the list of this animal name. If there’s no match, the function would not remove anything; if there is, then the function would remove the specific animal.

11. In the class, there is a function(called “exit”). In this function, release all memories that were dynamically allocated by the pointers. And only delete the memory when the corresponding pointer is pointing to something that’s not NULL.

12. There is a function that will be called in main, this function will display the welcome messages, explaining what this program can do.

13. There is a function that will be called in main and some of the member functions in the class, this function will ask if the user wants to do something again and prompt the user for an answer.

14. There is a function that will be called in main, this function takes an integer as argument (will be passed by reference) will display the menu and prompt the user for a number representing which task they want to do. In main, depends on the user’s input, specific code will be executed to complete specific missions.