Lab #5

Spring 2025

Requirements

In this lab, you will cover building and using a linked list. The linked list will be built out of Node structs, which are defined in the header file. This implementation will not use placeholder nodes in the list. All of your functions should assume that the list passed to it could be empty.

```
// lab5.h

typedef struct Node Node;

struct Node {
	int data;
	Node* next;
};
```

1.1 initList

```
Node* initList();
```

- **Info:** This function will return an empty list pointer.

1.2 insertAtTail

```
Node* insertAtTail(Node* head, int data, int* error);
```

- **Info:** This function will take a list pointer, an integer representing a list item, and an int pointer with which to return an error code. It will insert the given list item at the tail of the list in a newly created node. It will return a pointer to the head of the list after execution, and it will set error to 0 on success or 1 on failure.

1.3 removeFromHead

```
Node * removeFromHead(Node *head, int *data);
```

- **Info:** This function will take a list pointer and an int pointer with which to return the removed data item, and will remove the node at the head of the list. It will return NULL if the list is empty, otherwise a pointer to the head of the list.

1.4 printList

```
void printList (Node *head);
```

- **Info:** This function will take a list pointer, and print the elements on the list from head to tail.

1.5 listContains

```
int listContains(Node *head, int data);
```

- **Info:** This function will take a list pointer and an integer. It will return 1 if the given list contains the given integer, else 0.

1.6 freeList

```
void freeList(Node *head);
```

- **Info:** This function will take a list pointer and free all memory allocated to the list.

Submission Information

Submit this assignment by using the mucsmake command.

Use the following command on Hellbender:

mucsmake <course> <assignment> <filename>

For example:

mucsmake 2050 lab5 lab5.c

Rubric: 20 points

1. Write required *initialize* function
   * 1 point
2. Write required *insert* function
   * 7 points
3. Write required *remove* function
   * 5 points
4. Write required *print* function
   * 4 points
5. Write required *list contains* function
   * 2 points
6. Write required *free* function
   * 1 point

Notice:

1. All of your lab submissions **must** include documentation to receive full points.
2. All of your lab submissions must compile under GCC using the *−Wall* and *−Werror* flags to be considered for a grade.
3. You are expected to provide proper documentation in every lab submission, in the form of code comments. For an example of proper lab documentation and a clear description of our expectations, see the lab policy document.