Lab #5
Fall 2023

Requirements

In this lab, you will cover creating and maintaining linked lists. Your list implementation should **not** make use of any placeholder (or "dummy") nodes. The list pointer should always point to the **head** of the list. In this lab you are given the following struct definition:

```
struct Node {

        void * data ;
        Node * next;

};
```

1.1 makeList

```
int makeList(Node **list)
```

Info: This function will initialize the provided pointer to Node * to an empty list. It will **always** return 0.

1.2 getSize

```
int getSize(Node *list)
```

Info: This function takes a list, and returns the number of elements on the list. Note that if the list is empty, the number of elements on the list is 0.

1.3 insertAtHead

```
int insertAtHead(Node **list, void *data)
```

Info: This function takes a pointer to Node *, and inserts the given data at the head of the list. It returns 0 if insertion was successful, or 1 if insertion failed.

1.4 removeFromTail

```
void * removeFromTail(Node **list)
```

Info: This function takes a pointer to Node *, and removes the element at the tail of the list (if any). It will return the data which was removed from the list, or NULL if the list was empty. You may assume that no data on the list will be NULL.

1.5 freeList

```
void freeList(Node **list)
```

Info: This function takes a pointer to Node *, and frees the memory allocated to the list. After freeing, it sets the pointer to NULL. **Note that the data on the list is not considered part of the memory allocated to the list.**

Submission Information

Submit this assignment by using the mucsmake command.

Use the following submit command on tc.rnet:

mucsmake <assignment> <filename>

For example:

mucsmake lab5 lab5.c

Rubric: 18 points

1. Write required *makeList* function
   * 2 points
2. Write required *getSize* function
   * 3 points
3. Write required *insertAtHead* function
   * 5 points
4. Write required *removeFromTail* function
   * 5 points
5. Write required *freeList* function
   * 3 points

Notice:

1. All of your lab submissions **must** include documentation in the form of code comments to receive full points. In addition, your program is expected to have a **comment header** at the top that includes your name, pawprint, the course you are taking, and the lab that you solved.
2. All of your lab submissions must compile under GCC using the *–Wall* and *–Werror* flags (or alternatively, the ***compile*** command on tc.rnet.missouri.edu) to be considered for a grade.
3. Do NOT change the given function prototype or anything else in the provided .h file. #include statements (e.g. for required C libraries) are expected to go into the source file you submit.
4. In your submission, please reference the source of any code that was not created independently by yourself. For example, if you used code which was presented in class lectures, the source would be something like "CS 2050 Course Notes by Jim Ries".