Lab #7

Spring 2025

Requirements

In this lab, you will cover creating and maintaining a Stack ADT. You are only given a partial definition of the Stack implementation, and you must derive the remainder of the implementation from the given complexity requirements. Note that you may need to define extra helper functions or struct types to complete this lab, and that these extra definitions must go in your **lab7.c** file.

You need to implement struct _Stack in your lab7.c file. Notice that lab7.h only gives a "forward definition" of the type Stack.

1.1 initStack

```c
// O(1)
Stack * initStack()
```

Info: This function will initialize and return a Stack. If initialization fails, you will return NULL. Your grade for this function will also include your stack implementation.

1.2 getSize

```c
// O(1)
int getSize(Stack * s)
```

Info: This function takes a stack, and returns the number of elements on the stack.

1.3 peekStack

```c
// O(1)
void * peekStack(Stack * s)
```

Info: This function takes a stack, and returns the element at the top of the stack **without removing it**.

1.4 pushStack

```c
// O(1)
int pushStack(Stack * s, void *data)
```

Info: This function takes a stack, as well as a data item. It will push the item onto the top of the stack, and return 0 if insertion was successful, or 1 if it was not.

1.5 popStack

```
// O(1)
void * popStack(Stack * s)
```

Info: This function takes a stack, and pops the data item from the top of the stack. It returns the item popped from the stack, or NULL if the stack is empty.

1.6 stackContains

```
// O(n)

int stackContains(Stack * s, void *data)
```

Info: This function takes a stack, as well as a data item. It returns 1 if the given data exists on the stack, or 0 if it does not.

1.7 freeStack

```
// O(n)

void freeStack(Stack * s)
```

Info: This function takes a stack, and frees all memory allocated to the stack. **Remember that data which have been inserted onto the stack are not considered part of the memory allocated to the stack.**

Submission Information

Submit this assignment by using the mucsmake command.

Rubric: 20 points

1. Write required *initStack* function
   * 5 points
2. Write required *getSize* function
   * 2 points
3. Write required *peekStack* function
   * 2 points
4. Write required *pushStack* function
   * 2 points
5. Write required *popStack* function
   * 2 points
6. Write required *stackContains* function
   * 4 points
7. Write required *freeStack* function
   * 3 points

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.