

Class name:

We are implementing a class called Number_bucket that contains a dynamic sized LIFO(Last In First Out) data structure.

Class Functions;

The class Number_bucket will have the following functions:

1. isEmpty()
2. push(int n)
3. pop()
4. get_size()
5. top()

Description of the functions:

1. isEmpty() checks whether or not the stack is empty
2. push(int n) adds an integer into the stack.
3. pop() removes the integer from the top of the stack.
4. get_size() shows the number of elements currently added to the stack.
5. top() will return the int that is currently at the top of the stack.

Test cases for functions:

1. isEmpty() has two possible test cases:
 - a. Returns true if the stack is empty.
 - b. Returns false if the stack is not empty.
2. push(int n) has two possible test cases:
 - a. An int gets pushed into the stack
 - b. Nothing gets pushed to the stack
3. pop() has two possible test cases:
 - a. The element at top of the stack is removed.
 - b. Nothing gets removed from the stack.
4. get_size() has one possible test case:
 - a. An integer that is the size of the stack will be returned.
5. top() has two possible test cases:
 - a. An integer at the top of the stack will be returned.
 - b. Nothing gets returned from the stack.

Program Test Cases:

1. Test case for bucket 1: Adding numbers from the file to the bucket then outputting to a file.
2. Test case for bucket 2: Adding numbers from a file then popping only one number then outputting to a file.

3. Test case for bucket 3: Adding numbers from a file then popping all of them making the bucket empty then outputting to a file
4. Test case for bucket 4: Opening an empty input file then outputting.
5. Test case for bucket 5: Adding big amount of numbers from a file and adding numbers manually then performing a bunch of different operations then outputting the result.