Lab Test 2
CSE 4304
SWE B
08-06-2021

Instructions to solve the tasks:

1. Calculate the number of operations for solving the task for various inputs (n = 10, 100, 1000, 10000 and 100000). For simplicity consider each statement as one operation.
   For example, the following code segments need 17 operations to execute.

```
void testFunction(){
      int i, count = 0;
      count++;
      for (i = 0; i < 10; i++) {
            count++;
            if(i%2) {
                  printf ("%d ",i);
                  count++.
            }
      }
      count++;
      printf("\n Number of operations: %d\n", count);
}
```

   NB: Forgetting to count few constant numbers of operations are negligible. Make sure to count the operations which are impacted by the input size (i.e., operations inside loop).

2. Generate **n** random numbers and save them in a text file. Instead of reading the inputs from the keyboard, read the saved numbers from the text file as your input for the task.

   NB: Make sure the keep a copy the input text file and submit it along with your source code as a submission for the task.

Zero tolerance on plagiarism.
Both parties who share or copy the code of others will be equally penalized.

Compulsory task:

1. Declare an integer array and read **n** integer values from the text file (which is generated earlier) and store into the array one by one by calling a user defined function. Make sure the values are inserted in the array in ascending order. **Do not** sort the array later. **(40)**
    a. The insert function has two parameters. First parameter is the beginning address of the array, and second parameter is the value you want to insert in the array.
    b. The function does not return any value.

2. Write a user defined function which uses "binary search" to search for a given number inside an array which are passed as parameters to the function. **(40)**
    a. The function has two parameters. The first parameter is the beginning address of the array, and second parameter is the number you are searching for.
    b. The function returns an integer. If it finds the searching value, it returns the index of the array where it is found, otherwise it returns -1.

3. Count the number operations to complete the binary search for various input size (at least 6 different input size **n = 10, 100, 1000, 10000, 50000, 100000).** Put the numbers into a table and plot into a graph. **(20)**
   NB: You may use the sample excel file provided with the lab task to generate the graph.

Task1: (For the students having odd student ID)

1. Implement Queue using doubly linked list. **(50)**
   Hint: You may **NOT** need to implement all the functions of a doubly linked list. Only implement the functions necessary for **enqueue** and **dequeue** operation of Queue.
2. Write a user defined function to search for a data in the linked list. Return 1 if it is found, else return 0. Count the number of operations needed to search the number. **(50)**

Task2: (For the students having even students ID)

1. Implement Stack using double linked list. **(50)**
   Hint: You may **NOT** need to implement all the functions of a doubly linked list. Only implement the functions necessary for **push** and **pop** operation of Stack.
2. Using your implemented stack code, evaluate the value of a postfix expression. The postfix notation will be given as a form of a string, where each character represents either a digit or an operator. For simplicity, assume every number is a single digit number. **(50)**