National Institute of Technology, Calicut

Department of Computer Science and Engineering

CS2094D – Data Structures Lab

Assignment-3 (Advanced Batch)

Policies for Submission and Evaluation

You must submit your assignment in the moodle (Eduserver) course page, on or before the submission deadline. Also, ensure that your programs in the assignment must compile and execute without errors in Athena server. During evaluation your uploaded programs will be checked in Athena server only. Failure to execute programs in the assignment without compilation errors may lead to zero marks for that program. Your submission will also be tested for plagiarism, by automated tools. In case your code fails to pass the test, you will be straightaway awarded zero marks for this assignment and considered by the examiner for awarding F grade in the course. Detection of ANY malpractice regarding the lab course will also lead to awarding an F grade.

Naming Conventions for Submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>.zip For example: ASSG4_BxxyyyyCS_LAXMAN.zip). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive. The source codes must be named as ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>_<PROGRAM-NUMBER>.<extension> (For example: ASSG4_BxxyyyyCS_LAXMAN_1.c). If there is a part a and a part b or a particular question, then name the source files for each part separately as inASSG4_BxxyyyyCS_LAXMAN_1b.c.

If you do not conform to the above naming conventions, your submission might not be recognized by some automated tools, and hence will lead to a score of 0 for the submission. So, make sure that you follow the naming conventions.Standard of Conduct Violations of academic integrity will be severely penalized.

Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course.

The department policy on academic integrity can be found at:
http://cse.nitc.ac.in/sites/default/files/Academic-Integrity.pdf .

Assignment Questions

Questions 1-4 are in the common questions for Main and Advanced Batches

5. Implement a C program that uses dynamic arrays for implementing the following operations. Dynamic arrays are arrays that can grow depending upon the number of elements in the array. (A similar concept to the dynamic tables discussed in class). Initial size of the array need not be specified. The amortized complexity of each operation is also mentioned in the following table along with the description of each function.

Let A={1,2,3} and B={4,5}, x=4, a=0,b=1

Function	**Description**	**Amortized Cost**	**Output**
Append(A, x)	Appends the element x at the end of the array.	O(1)	(1 2 3 4)
Concatenate(A,B)	Concatenates the array B at the end of array A. Array B remains unchanged.	O(n+n)	(1 2 3 4 5)

Member(A, x)	Returns 1 if x is member of A and otherwise 0.	O(n)	0
Length(A)	Returns the length of array A	O(1)	3
Slice(A,a,b)	Returns the elements of array A from the index a-1 to b-1.	O(k),k=b-a	(1,2)

Input format

Your program should read input from **input.txt** and write the output to **output.txt**
The input consists of multiple lines, each one containing a character from {'a', 'c', 'm','l','s','x'} followed by zero, one or two integers or a set of integers that represents an array of elements. The integer(s), if given, is in the range 0 to 10000. In the input a dynamic array is given a set of integers inside a pair of parenthesis.

● Call the function append(A,x) if the input line contains the character 'a' followed by an integer array and an integer x .
Output the resultant array A after appending the element x at the end of the array.

● Call the function concatenate(A,B) if the input line contains the character 'c' followed by two sets of integers.
Output the resultant array after concatenating A and B.

● Call the function member(A,x) if the input line contains the character 'm' followed by space separated a set of integers A and an integer x . Print **0**, without terminating, if x is not present in the array A. Otherwise, print **1**.

● Call the function length(A) if the input line contains the character 'l' find the number of elements in the array and print the value.

● If the input line contains the character 's' followed by an array A and two integers x and b, print the elements of A from the index a-1 to b-1.

● If the input line contains the character 'x' then exit.

Output format

The output (if any) of each command should be printed on a separate line.

Sample Input:

```
a (1 2 3) 4
c (1 2 3) (2 4 5)
m (3 5 7 9) 9
m (1 2 3) 6
l (1 2 3 4 5 6)
```

s (1 2 3 4 5 6 7) 3 6
x

Sample Output
(1 2 3 4)
(1 2 3 2 4 5)
1
0
6
3 4 5 6

6. Write a C program to implement dynamic Binary Search that supports insertion, deletion and search operations. The amortized running time of the operations should be as given below:

Insertion- O(log n), Search - O(log n)

[**Dynamic Binary Search** : Binary search of a sorted array takes logarithmic search time, but the time to insert a new element is linear in the size of the array. We can improve the time for insertion by keeping several sorted arrays.

Specifically, suppose that we wish to support SEARCH and INSERT on a set of n elements. Let $k = \lceil \lg(n+1) \rceil$, and let the binary representation of n be $\langle n_{k-1} n_{k-2},\ldots,n_0$. We have k sorted arrays $A_0,A_1,\ldots,A_{k-1}$ where for $i=0,1,\ldots,k-1$, the length of array A_i is 2^i. Each array is either full or empty, depending on whether $n_i =1$ or $n_i=0$, respectively. The total number of elements held in all k arrays is therefore $\sum_{i=0}^{k-1} n_i$. Although each individual array is sorted, elements in different arrays bear no particular relationship to each other.]

Input format
Your program should read input from **input.txt** and write the output to **output.txt**
The input consists of multiple lines, each one containing a character from {'i', 's', 'p','x'}followed by zero or one integers The integer(s), if given, is in the range 0 to 10000.

● Call the function insert(A,x) if the input line contains the character 'i' followed by an integer array and an integer x

● Call the function search(A,x) if the input line contains the character 's' followed by an integer x. Print the resultant index of x in the array if x is in the array and otherwise print **NIL**.

● Call the function print(A) to print the contents of the array A if the input line contains the character 'p' . Elements in non empty arrays are printed as $A_0, A_1, \ldots, A_{k-1}$. Elements are separated by single white space.

● If the input line contains the character 'x' then exit.

Output format
The output (if any) of each command should be printed on a separate line.

Sample Input

```
i  4
i  2
p
i  3
p
s  2
i 1
p
s 5
i  7
p
x
```

Sample Output

```
2 4
3 2 4
1
1 2 3 4
NIL
7 1 2 3 4
```