

**Birla Institute of Technology & Science, Pilani**  
**2<sup>nd</sup> Semester 2016-17 - CS F211 – Data Structures and Algorithms**

---

**Lab 11 (Evaluation 3): 22<sup>nd</sup> April, 2017**

**Time: 170 minutes**

**Marks: 20 + 10 = 30**

**Instructions:**

- *This test consists of two problems (Problem 1 and Problem 2) specified in two different files.*
- All input expressions should be read from stdin (scanf) and output should be printed on stdout (printf).
- For first 150 minutes, only a subset of test cases will be visible to students after submitting the code on the portal. Only in last 20 minutes, all test cases will be made visible.
- At the end of 170 minute period, the online system will stop evaluating the submissions but it will accept it for additional 10 minutes. At the end of 180 minute period, it will stop accepting the submissions.
- Only the last submission by the student for each problem will be considered for evaluation, irrespective of earlier correct submission.
- Assuming that a problem contains M marks, in case of (Run-error/Compiler-error/Timelimit-error), evaluation will be done for M/2 marks only.
- Total marks of each problem contains some marks for modularity and proper structuring of code.
- All submitted source code will be later checked manually by the instructor and final marks will be awarded. Any case of plagiarism and/or hard coding of test cases will fetch 0 marks for the problem/evaluation component.
- Make sure to return 0 from the main() function in case of normal termination.

## Problem 1 of 2

**Expected Time: 120 minutes**

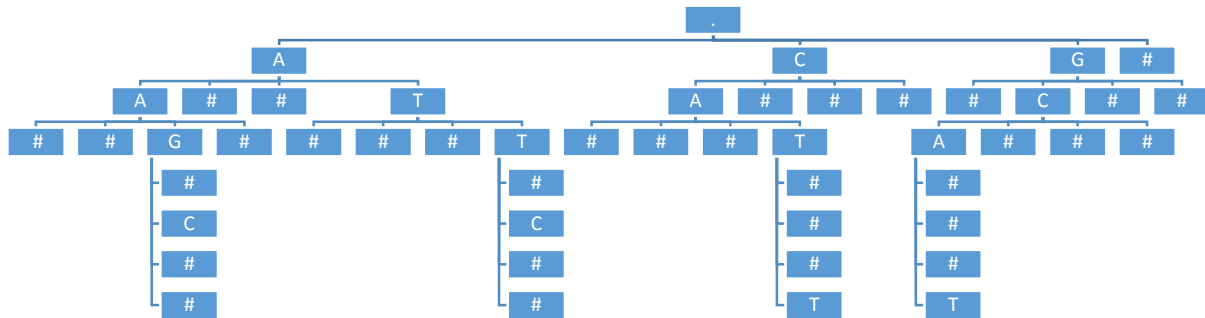
**Marks: 20**

**Problem Statement**

Write a procedure that will construct a *trie* from a given set of DNA sequences by repeated insertion. Each DNA sequence is a sequence of proteins marked by characters A, C, G, or T.

Root will contain a dot (.) character as its value.

For instance, given the set {AAGC, ATTC, CATT, GCAT} the following *trie* would be constructed (# denotes empty locations in diagram):



### Input format

Each line will start with a one of the following key values (0, 1, 2, 3, -1). Each key corresponds to a function and has a pattern. Implement following function according to given pattern and write a driver function which can call the functions according to given key value.

Key	Function to call	Format	Description
0	createTrie	0 sequence	"0" shows creation of a trie. "sequence" will be a set of strings (words) of upper case characters (A, C, G, or T), separated by space. Character (X) will represent termination of sequence.
1	printTrie	1	Print the trie in Depth First Order (in order of A, C, G, T at same level), separated by space, in a new line.
2	find	2 str	2 shows call to a find operation which takes an input string <i>str</i> searches for it in trie and prints "1" if <i>str</i> is present, "-1" otherwise.
3	findPrefix	3 pre	Implement a findPrefix operation which takes an input string <i>pre</i> and searches for a subtrie <i>S</i> such that for any string <i>suf</i> in <i>S</i> , <i>pre</i> concatenated with <i>suf</i> is present in the trie. Call printTrie on subtrie S.
-1	stop		stop the program

### Test Case 1:

Input	Output
0 AAGC ATTC CATT GCAT X 1 2 CAT 2 ACT 3 A -1	A A G C T T C C A T T G C A T 1 -1 A A G C T T C

**Test Case 2:**

**Input →**

```
0
CAAGC CAAGA CATT CCATT CGCAT AAAGC AAAGA AATTC ACATT AGCAT X
1
2 CAA
3 AA
3 A
-1
```

**Output →**

```
A A A G A C T T C C A T T G C A T C A A G A C T T C C A T T G C A T
1
A A G A C T T C
A A A G A C T T C C A T T G C A T
```