IIT PALAKKAD

COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology, Palakkad
CS2130: Data Structures and Algorithms Lab
Lab 13 (Mock Exam)

9 Nov, 2018

Time: 3 hrs

General Instructions

- You are expected to follow good programming practices (refer to the book titled *C How to Program* by Dietel and Dietel). Your programs should handle all kinds of test cases - positive, negative, corner cases/boundary conditions, good input, bad input and are never expected to crash on any input.
- You are also expected to provided a high level description of the functions used in your programs. There is no restriction on the format except that it should be concise and precise. Further, the running time and additional space (excluding the space for the arguments) used by the functions should be clearly stated and justified.
- **Please note that poor programming practices and improper documentation can attract a penalty of up to 20% of the marks allocated to a question.**

1. (45 points) Write a program to convert an arithmetic expression given in postfix form into its infix form.

   Input-Output Format: The input consists of a single line containing the input postfix expression. Assume that the input is a valid postfix expression where each operand is represented by a single lower case letter and each of the operators come from the set $\{+, -, *, /\}$. The output consists of a single line containing the infix expression without brackets. No additional newline is required in the output.

   ```
   Sample Input:
   abcd-*+efg*++h-

   Expected Output:
   a+b*c-d+e+f*g-h
   ```

2. (30 points) Recall that the distance from vertex **s** to vertex **t** is the length of a shortest path from **s** to **t**; if there is no path from **s** to **t** then the distance is -1. Given a directed graph **G** and two vertices **s** and **t**, write a program to find the distance from **s** to **t**.

   Input. The input is a directed graph **G** on **n** vertices and **m** edges where $\mathbf{n} \geq \mathbf{1}$. The vertex set of **G** is $\{0, 1, \ldots, n-1\}$ and an edge directed from vertex **i** to vertex **j** is denoted by **(i, j)**. The input file will contain **m+2** lines with the first line containing the number **n** and the second line containing two vertices **s** and **t** of **G** separated by a space. Each of the subsequent **m** lines will contain two non-negative integers **i** and **j**, separated by a space, denoting the edge

Continued on next page...

(i, j) of **G**.

Output. The output consists of a single line containing the distance from vertex **s** to vertex **t** displayed in the format given below. No additional newline is required in the output.

```
Sample Input:
10
4 9
0 1
0 5
0 2
1 4
2 3
4 5
3 0
4 7
3 7
4 6
5 6
7 6
7 8
8 9
5 1

Expected Output:
dist(4,9)=3
```

3. (25 points) Consider the following scenario of job scheduling on a single-processor machine. Each job has an identifier **id** denoted by a single character and a priority **pr** denoted by a non-negative integer. A job with priority value **r** has higher priority than any job with priority value strictly lesser than **r**. At any point of time, there is a set (not necessarily non-empty) **S** of jobs that are required to be executed. Not all jobs are to be executed as they arrive and any new job that arrives is added to the current set **S** of available jobs. Write a program to implement the following standard operations in this scenario. Each of the these operations should be performed in $\mathcal{O}(\log |\mathbf{S}|)$ time. You may assume that at any point of time, the jobs under consideration have distinct priorities and distinct identifiers.

   ADD jobID jobPri. Add the job with identifier **jobID** and priority **pri** into the current set **S** of jobs. **jobID** is a character and belongs to the set $\{a, b, \ldots, z\} \cup \{A, B, \ldots, Z\}$ Output -1 if the operation is unsuccessful.
   EXE. Execute the job in **S** that has maximum priority. Output the identifier of the job executed and delete this job from **S**. If no such job exists, output -1.

   Input-Output Format: The input will be given in a file where the first line contains the value of **n**, the maximum size of the set **S** of available jobs at any point of time. The subsequent lines will contain specific operations on **S**.

Question 3 continues on the next page…

```
Sample Input:
10
ADD P 15
ADD Q 26
ADD R 33
ADD S 16
EXE
EXE
EXE
EXE
EXE
EXE
EXE

Expected Output:
R
Q
S
P
-1
-1
-1
```