National Institute of Technology, Calicut

Department of Computer Science and Engineering

CS2094 – 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: ASSG3_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: ASSG3_BxxyyyyCS_LAXMAN_1.c). If there is a part *a* and a part *b* for a particular question, then name the source files for each part separately as in ASSG3_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

1. Write a program to implement reversal of a Singly Linked List in O(1) additional space.

Input format :
The input contains space separated integers in the range -2^{30} to 2^{30}, the contents of the linked list.

Output format :
Space separated integers, representing the reversed linked list.

Sample Input:
2 4 3 1 9
Sample Output:
9 1 3 4 2

2. An n-digit (where, 1 <= n <= 10000) positive integer is represented by a singly linked list (SLL) of length *n* in the following way:

Each digit of the number is stored in a node of SLL so that the most significant digit is at the node pointed to by the head pointer of the linked list and the least significant digit is at the tail of SLL.
For example, the number 673924 is represented as 6 -> 7 -> 3 -> 9 -> 2 -> 4.

Write a recursive function **difference_of_digits()** that takes as input two SLLs representing two n-digit numbers as arguments and subtracts the smallest SLL from the largest SLL and stores the difference between both the input SLLs in the largest SLL. The function should only use O(1) additional space and should run in O(n) time.

Input format :
First two lines contain *n-digit* positive integers.

Output format :
An *n* or <*n* digit number, containing the difference of the input numbers

Sample Input :
654321
532187

Sample Output :
122134

3. Write a program to implement a queue using the following:
   a) Array representation
   b) Linked list representation

Your program must support the following functions:

1. **enqueue(element) -** Puts the data specified by **element** at the rear end of the queue**.** Return -2, if the queue is full.
2. **dequeue** - removes and returns the element at the front of the queue. Return -1, if the queue is empty.
3. **print_queue** - displays the values of all the elements present in the queue.

Input format:
The input consists of multiple lines, each line of the input contains a character from {'e', 'd', 'p', 's'} followed by zero or one integer. The integer, if given, is in the range 0 to 2^{31}.

- The character 'e' means enqueue the next integer from the input into the queue. In this case, the next integer (>= 0) is given on the same line as the character 'e', separated by a space.

- The character 'd' means dequeue and output element from the front of the queue. Output “-1”, if the queue was originally empty.

- The character 'p' means show all elements in the queue. In this case, output all elements of the queue on a single line, separated by space, starting with the element at the front. Output “-1”, if the queue was originally empty.

- The character 's' means “stop the program”

Output format:

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

Example:

a) *Array Representation*: For the array implementation, the very first line of input contains an integer value c, 0 < c < 100, which is the capacity of the queue. In this case, the enqueue operation must output “-2” when an element is being enqueued into an already full queue.

Sample Input:

```
3
e 2
e 5
e 7
d
p
e 10
e 11
d
p
d
d
d
e 3
p
s
```

Sample Output:

```
2
5 7
-2
5
7 10
7
10
-1
3
```

b) *Linked list representation*

Sample Input:

```
e 2
e 5
```

```
e 7
d
p
e 10
d
p
d
d
d
e 3
p
s
```

Sample Output:

```
2
5 7
5
7 10
7
10
-1
3
```

4. Write a program to implement a priority queue using heap. Your program must support the following functions:

1. **insert(pq, elem, pr) -** inserts the data specified by element **elem** into the priority queue **pq** with priority **pr**.

2. **extract_min(pq) -** removes and prints the element with the lowest priority from the priority queue **pq**; prints “EMPTY” if the priority queue is empty. (Strictly follow the output format. It should be EMPTY not empty or anything else.)

3. **get_min(pq) -** prints the element with the lowest priority from the priority queue **pq**, without actually removing it from the priority queue; prints “EMPTY” if the priority queue is empty. (Strictly follow the output format. It should be EMPTY not empty or anything else.)

4. **decrease_priority(pq, elem, newpr) -** changes the priority of element **elem** in the priority queue specified by **pq**, by assigning it the new priority **newpr**. It is guaranteed that **elem** will be present in the priority queue, and that **newpr** will be lower (in the sense of priority) than the original priority of **elem**.

Input format:

- The input consists of multiple lines, each line of the input contains a character from {'a', 'e', 'g', 'd', 's'} followed by zero or two integers. The integers, if given, are in the range -2^{30} to 2^{30}.
- The character 'a' means insert the next integer from the input into the priority queue. In this case, two more integers follow the character 'a', each separated by space. The first integer (>= 0) is the data to be inserted. The second integer (>= 1) is the priority of this data item (1 being the lowest priority). Assign the priority to the data, and then insert this data into the priority queue.

- The character 'e' means remove and output the element with the lowest priority from the priority queue (Output the priority of the element in parenthesis, separated by a space). Output “EMPTY”, if the priority queue was originally empty. (Strictly follow the output format. It should be EMPTY not empty or anything else.)

- The character 'g' means output the element with the lowest priority from the priority queue without actually removing it (Output the priority of the element in parenthesis, separated by a space). Output “EMPTY”, if the priority queue was originally empty. (Strictly follow the output format. It should be EMPTY not empty or anything else.)

- The character 'd' means decrease the priority of a datum in the priority queue. In this case, two more integers follow 'd', each separated by space. The first integer (>= 0) is the data (which will be present in the priority queue). The second integer (>= 1) is the new decreased priority. After this operation, the priority of this data should be the one specified in this input.

- The character 's' means “stop the program”.

Output format:

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

Sample Input:

```
a 2 100
a 5 110
a 7 95
e
```

```
e
a 10 120
e
a 6 30
g
d 10 25
g
e
e
e
a 1 30
a 2 15
e
s
```

Sample Output:

```
7 (95)
2 (100)
5 (110)
6 (30)
10 (25)
10 (25)
6 (30)
EMPTY
2 (15)
```

5. Write a program to convert the Postfix expression to Infix expression and print the result of the evaluated expression.

Input format :

The input consists of Postfix expression in which each symbol is separated by a single space.

Output format :

The first line prints the Infix expression corresponding to the input **without any space**.
The second line, prints the result of the expression

Sample Input:

```
10 5 2 * + 6 3 / 1 * -
```

Sample Output:

```
10+5*2–6/3*1
18
```