National Institute of Technology Calicut
Department of Computer Science and Engineering
Fourth Semester B. Tech.(CSE)-Winter 2023-24
CS2094D Data Structures Laboratory
Assignment Cycle#1
Part A

Submission deadline (on or before): 18.01.2024, 11:00 PM

Policies for Submission and Evaluation:

- You must submit your assignment in the Eduserver course page, on or before the submission deadline.
- Ensure that your programs will compile and execute without errors using gcc compiler.
- During the evaluation, failure to execute programs without compilation errors may lead to zero marks for that evaluation.
- 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 related to the lab course can lead to awarding an F grade in the course.

Naming Conventions for Submission

- Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar, .tar, .gz). The name of this file must be

  `ASSGC<NUMBER>_<PART>_<ROLLNO>_<BATCHNO>_<FIRST-NAME>.zip`

  (Example: *ASSGC1_A_BxxyyyyCS_CS01_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

  `ASSGC<NUMBER>_<PART>_<ROLLNO>_<BATCHNO>_<FIRST-NAME>_<PROGRAM-NUMBER>.c`

  (For example: *ASSGC1_A_BxxyyyyCS_CS01_LAXMAN_1.c*). If you do not conform to the above naming conventions, your submission might not be recognized by our automated tools, and hence will lead to a score of 0 marks for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

- Violation 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: https://minerva.nitc.ac.in/?q=node/650.

QUESTIONS

1. You have given mathematical expressions which have some duplicate parentheses. The expressions consist of lowercase alphabets and operators (+, -, *, /, ˆ , %) without space, Your goal is to identify and flag expressions that have duplicate parentheses. Duplicate parentheses occur when the same subexpression is enclosed by multiple sets of parentheses. It's important to note that a single alphabet or operator by itself is also considered a valid expression. Hint: use stack

   Input format:

   - The first line is an integer where $n \in [1, 10^6]$.
   - The second line contains valid mathematical expressions of size 'n'

   Output format:

   - The output consists of a single line with an integer value (0 or 1)
     0 representing the absence of duplicate parentheses
     1 representing the presence of duplicate parentheses

   Sample Input 1:
   15
   (a+b)%(c+(a-b))

   Sample Output 1:
   0

   Sample Input 2:
   7
   (a+(b))

   Sample Output 2:
   1

2. A Singly linked list L is a data structure in which the objects are arranged in a linear order. Each node of a Singly linked list L is an object with an attribute *key* and one pointer attribute, *next*. Given a node x in the list, *x.next* points to its successor in the linked list. An attribute *L.head* points to the first node of the list.

   Write a menu driven program to implement an unsorted Singly linked list L. Your program must contain the following functions:

   Your main() function shall read the choice from the console and call the following functions appropriately: Your program must contain the following functions:

   - Main() - repeatedly reads a character '*a*', '*r*', '*d*', '*p*', '*s*' from the terminal and calls the sub-functions appropriately until character '*e*' is entered.
   - CREATE-NODE(k) - creates a new node with *key* k and pointer *next* of node points to NULL. This procedure returns a pointer to the new node.
   - LIST-INSERT(L, x) - inserts x as the last node of L.
   - LIST-DELETE(L, i) - deletes the node at index i from L and print the deleted key value. The position is determined by its index, considering first node at index 1. If i is not found in L, then print -1
   - LIST-Rduplicate(L) - Remove duplicates: This function will remove duplicates from the list L and after removal the updated list L is printed.
   - LIST-Pal(L): Function is used to check whether the linkedlist L is palindrome or not.The function will return Y (if it is a palindrome) otherwise N.
     (In case of even length linkedlist , the first node is considered as middle node).

- LIST-DISPLAY(L) - prints the list L

Note:- For every INSERT operation, the node x is created by calling CREATE-NODE() function.

Input format:

- Each line contains a character from '*a*', '*r*', '*d*', '*p*', '*s*', '*e*' followed by zero, one or two integers. The integers, if given, are in the range $[-10^6, 10^6]$.
- Character '*a*' is followed by an integer separated by space. In this operation, the node with this integer as key is inserted to the tail of L.
- Character '*r*' is followed by an integer i separated by space. In this operation, the node at index i is deleted from L and the deleted node's key value is printed. Considering first node at index 1, if i is not found in L, then print -1
- Character '*d*' This operation will remove duplicates from the list L and after removal the updated list L is printed.
- Character '*p*' is f to check whether the linkedlist L is palindrome or not.The function will return Y (if it is a palindrome) otherwise N.
  (In case of even length linkedlist , the first node is considered as middle node).
- Character '*s*' This operation is to print the list L.
- Character '*e*' is to 'exit' from the program.

Output Format:

- The output (if any) of each command should be printed on a separate line.
- For options '*r*', '*d*' , '*p*' and '*s*' if a node with the input key is not present in L or L is empty, then print -1.

Sample Input 1:
```
a 1
a 2
a 2
a 1
a 5
a 6
r 6
r 5
p
d
s
```

Sample Output 1:
```
6
5
Y
1 2
1 2
```

Sample Input 2:
```
a 1
a 3
a 7
a 8
a 2
```

```
a 5
a 3
r 2
r 3
p
d
s
```

Sample Output 2:

```
3
8
N
1 7 2 5 3
1 7 2 5 3
```

3. You are responsible for designing a system to manage an emergency room (ER) using a priority queue(max heap). Patients arrive at the ER and are characterized by their **name**, condition severity (**priority**), and admit time(**admitTime**). The system needs to support the following functions:

   - **main()**: Repeatedly reads an input character from the menu list through the terminal and execute menu driven operations accordingly.The Menu list is [‘a’,’t’,’c’,’d’,’u’,’p’,’e’].The program ends on input ‘e’.
   - **admitPatient(priorityQ,name,priority,admitTime)**: Add a new patient to the ER with their name, condition severity(priority), and admit time.
   - **treatNextPatient(priorityQ)**: Treat the next patient in the ER with high severity. If multiple patients have the same severity, treat the one who was admitted first. Discharges the patient after treatment.
   - **checkNextPatient(priorityQ)**: Print the details(space separated) of the next patient to be treated without removing them from the priority queue.
   - **dischargePatient(priorityQ,name,admitTime)**: Discharge a specific patient from the ER based on their name and admitTime.
   - **updateConditionSeverity(priorityQ,name,admitTime, newPriority)**: Update the condition severity of a specific patient currently in the ER. This operation may change the patient’s position in the priority queue.
   - **printAllPatients(priorityQ)**: Print the details (space separated) of all patients (each patient in a new line) in the priority order. Consider admitting time for patients with the same priority.

   Input format:

   - Each line of input contains two parts separated by a space; the first part is a character from the menu list [’a’,’t’,’c’,’d’,’u’,’p’,’e’], and the second part contains parameters (space separated) according to the menu option.
   - Input ‘a’ followed by patient name (string), priority (int), and admit time (string) calls the function admitPatient(priorityQ,name,priority,admitTime), which pushes a new node containing these patient details into the priority queue.
   - Input ‘t’ calls the function treatNextPatient(priorityQ), which prints(space separated) and removes the patient node with high priority.
   - Input ‘c’ calls the function checkNextPatient(priorityQ), which prints(space separated) the patient with high priority.
   - Input ‘d’ followed by patient name and admit time calls the function dischargePatient(priorityQ, name,admitTime), which removes the specific patient node from the priority queue.

- Input ‘u’ followed by patient name, admit time, and new priority calls the function updateConditionSeverity(priorityQ,name,admitTime, newPriority), which updates the priority of the specific patient node.
- Input ‘p’ calls the function printAllPatients(priorityQ), which prints the details (space separated) of all the patients, each patient in a new line following the priority order.
- Input ‘e’ terminates the execution of the program.

Output format:

- Line contains string(patient name), integer(priority) and string(admit time).

Sample Input 1:
```
a John 3 08:30
a Mary 2 08:45
a Bob 4 09:00
p
t
p
d Mary 08:45
p
a Ramesh 5 11:30
u John 08:30 6
c
p
e
```

Sample Output 1 :
```
Bob 4 09:00
John 3 08:30
Mary 2 08:45
Bob 4 09:00
John 3 08:30
Mary 2 08:45
John 3 08:30
John 6 08:30
John 6 08:30
Ramesh 5 11:30
```

Sample Input 2:
```
a James 7 10:30
a Kumar 2 11:27
a Chitra 3 12:00
a Ahmed 9 14:00
a Kunal 7 18:00
p
t
d Kumar 11:27
p
c
u Chitra 12:00 9
d Kunal 18:00
p
e
```

Sample Output 2 :

```
Ahmed 9 14:00
James 7 10:30
Kunal 7 18:00
Chitra 3 12:00
Kumar 2 11:27
Ahmed 9 14:00
James 7 10:30
Kunal 7 18:00
Chitra 3 12:00
James 7 10:30
Chitra 9 12:00
James 7 10:30
```