



# Birla Institute of Technology & Science, Pilani

Work-Integrated Learning Programmes Division

MTech in Data Science & Engineering

S1\_2022-2023, DSECLZG519- Data Structures & Algorithms Design

## Assignment 1 – PS14 - Name Chain- [Weightage 12%]

*Read through this entire document very carefully before you start!*

### 1. Problem Statement

You have to store the names of the people standing in the line using a Doubly Linked List where every node represents a single person.

You need to allow the following functionality:

1. `add_at_start::Dave` > This will add Dave at the start of the line
2. `add_at_end::Ricky` > This will add Ricky at the end of the line
3. `add_at_pos::2::Harry` > This will add Harry to the line at 2nd position (The starting position is considered as first, if the position is outside the list add at the end)
4. `remove_people::2::1` > This will remove 2 people from the line starting from 1st position (You can assume that you will always be able to remove these people)
5. `flip_order::2::3` > Flip 2 people starting from 3rd position

### Requirements

1. Implement the above problem statement using Doubly linked list Data Structure and Python 3.7
2. Read the input from a file **inputPS14.txt**
3. You will output your answers to a file **outputPS14.txt**
4. Perform an analysis for the features above and give the running time in terms of input size: n.

### Example Input:

`add_at_start::Dave`

```

add_at_end::Ricky
add_at_pos::2::Harry
add_at_pos::7::Adam
add_at_pos::1::Jerry
add_at_pos::2::Rob
flip_order::2::3
remove_people::2::

```

### Example Output:

Space separated representation of the current line after each command

```

Dave
Dave Ricky
Dave Harry Ricky
Dave Harry Ricky Adam Jerry
Dave Harry Ricky Adam
Jerry Rob Dave Harry Ricky Adam
Jerry Rob Harry Dave Ricky Adam
Harry Dave Ricky Adam

```

### Explanation:

Input	Current Line					
Starting is Empty						
add_at_start::Dave	Dave					
add_at_end::Ricky	Dave	Ricky				
add_at_pos::2::Harry	Dave	Harry	Ricky			
add_at_pos::7::Adam	Dave	Harry	Ricky	Adam		
add_at_pos::1::Jerry	Jerry	Dave	Harry	Ricky	Adam	
add_at_pos::2::Rob	Jerry	Rob	Dave	Harry	Ricky	Adam
flip_order::2::3	Jerry	Rob	Harry	Dave	Ricky	Adam
remove_people::2::1	Harry	Dave	Ricky	Adam		

Remember: The starting position is 1

```
# Doubly linked list Data Structure (node)
class Node:
    def __init__(self, data=None, next=None, prev=None):
        self.next = next # reference to next node in DLL
        self.prev = prev # reference to previous node in DLL
        self.data = data
```

## Sample Files

### Sample Input:

Input will be taken from the file(**inputPS14.txt**) here each line is a command that needs to be executed

```
add_at_start::Dave
add_at_end::Ricky
add_at_pos::2::Harry
add_at_pos::7::Adam
add_at_pos::1::Jerry
add_at_pos::2::Rob
flip_order::2::3
remove_people::2::1
```

### Sample Output:

For every *line in the input* find the output the Space separated representation of the current line after each command

```
Dave
Dave Ricky
Dave Harry Ricky
Dave Harry Ricky Adam Jerry
Dave Harry Ricky Adam
Jerry Rob Dave Harry Ricky Adam
Jerry Rob Harry Dave Ricky Adam
Harry Dave Ricky Adam
```

Display the output in **outputPS14.txt**.

***Note that the input/output data shown here is only for understanding and testing, the actual file used for evaluation will be different.***

## 2. Deliverables

1. PDF document **designPS14\_<group id>.pdf** detailing your design approach and time complexity of the algorithm and alternate solutions.
2. **[Group id] \_Contribution.xlsx** mentioning the contribution of each student in terms of percentage of work done. Columns must be "Student Registration Number", "Name", "Percentage of contribution out of 100%". If a student did not contribute at all, it will be 0%, if all contributed then 100% for all.
3. **inputPS14.txt** file used for testing
4. **outputPS14.txt** file generated while testing
5. **.py file** containing the python code. Create a single \*.py file for code. Do not fragment your code into multiple files.
6. **Zip all of the above files including the design document and contribution file in a folder with the name: [Group id]\_A1\_PS14.zip** and submit the zipped file in canvas.
7. **Group Id** should be given as **Gxx** where xx is your group number. For example, if your group is 26, then you will enter G26 as your group id.

## 3. Instructions

1. It is compulsory to make use of the data structure(s) / algorithms mentioned in the problem statement.
2. Ensure that all data structure insert and delete operations throw appropriate messages when their capacity is empty or full. Also ensure basic error handling is implemented.
3. For the purposes of testing, you may implement some functions to print the data structures or other test data. But all such functions must be commented before submission.
4. Make sure that you read, understand, and follow all the instructions
5. Ensure that the input, prompt and output file guidelines are adhered to. Deviations from the mentioned formats will not be entertained.
6. The input, prompt and output samples shown here are only a representation of the syntax to be used. Actual files used to evaluate the submissions will be different. Hence, do not hard code any values into the code.
7. Run time analysis is to be provided in asymptotic notations and not timestamp based runtimes in sec or milliseconds.

8. Please note that the design document must include:
  - a. The data structure model you chose with justifications
  - b. Details of each operations with the time complexity and reasons why the chosen operations are efficient for the given representation
  - c. One alternate way of modeling the problem with the cost implications.
9. Writing a good technical report and well documented code is an art. Your report cannot exceed 4 pages. Your code must be modular and quite well documented.
10. You may ask queries in the dedicated [discussion section](#). Beware that only hints will be provided and queries asked **in other channels will not be responded to**.

#### **Instructions for use of Python:**

1. Implement the above problem statement using Python 3.7+.
2. Use only native data types like lists and tuples in Python, do not use dictionaries provided in Python. Use of external libraries like graph, numpy, pandas library etc. is not allowed. The purpose of the assignment is for you to learn how these data structures are constructed and how they work internally.
3. Create a single \*.py file for code. Do not fragment your code into multiple files.
4. Do not submit a Jupyter Notebook (no \*.ipynb). These submissions will not be evaluated.
5. Read the input file and create the output file in the root folder itself along with your .py file. Do not create separate folders for input and output files.

#### **4. Deadline**

1. The strict deadline for submission of the assignment is **Friday, 30th Dec 2022 11:55PM.**
2. The deadline has been set considering extra days from the regular duration in order to accommodate any challenges you might face. No further extensions will be entertained.
3. Late submissions will not be evaluated.

#### **5. How to submit**

1. This is a group assignment.

2. Each group has to make one submission (only one, no resubmission) of solutions.
3. Each group should zip all the deliverables in one zip file and name the zipped file as mentioned above.
4. Assignments should be submitted via Canvas > Assignment section. Assignments submitted via other means like email etc. will not be graded.

## 6. Evaluation

1. The assignment carries **12 Marks**.
2. Grading will depend on:
  - a. Fully executable code with all functionality working as expected
  - b. Well-structured and commented code
  - c. Accuracy of the run time analysis and design document.
  - d. Every bug in the functionality will have negative marking.
  - e. Marks will be deducted if your program fails to read the input file used for evaluation due to change / deviation from the required syntax.
  - f. Use of only native data types and avoiding libraries like numpy, graph and pandas will get additional marks.
3. We encourage students to take the upcoming assignments and examinations seriously and submit only original work. Please note that plagiarism in assignments will be taken seriously. All groups that are booked under plagiarism will be given 0 marks and no further discussion will be entertained. Please refer to the detailed policy [here](#).
4. Source code files which contain compilation errors will get at most 25% of the value of that question.

## 7. Readings

**Text book:** Algorithms Design: Foundations, Analysis and Internet Examples  
Michael T. Goodrich, Roberto Tamassia, 2006, Wiley (Students Edition). **Chapters on linked list.**

