**

***LAB # 08***

Implementing singly linked list, associated operations and Runner technique and Implementing doubly linked list, associated operations and LRU technique.

November 25, 2024

**

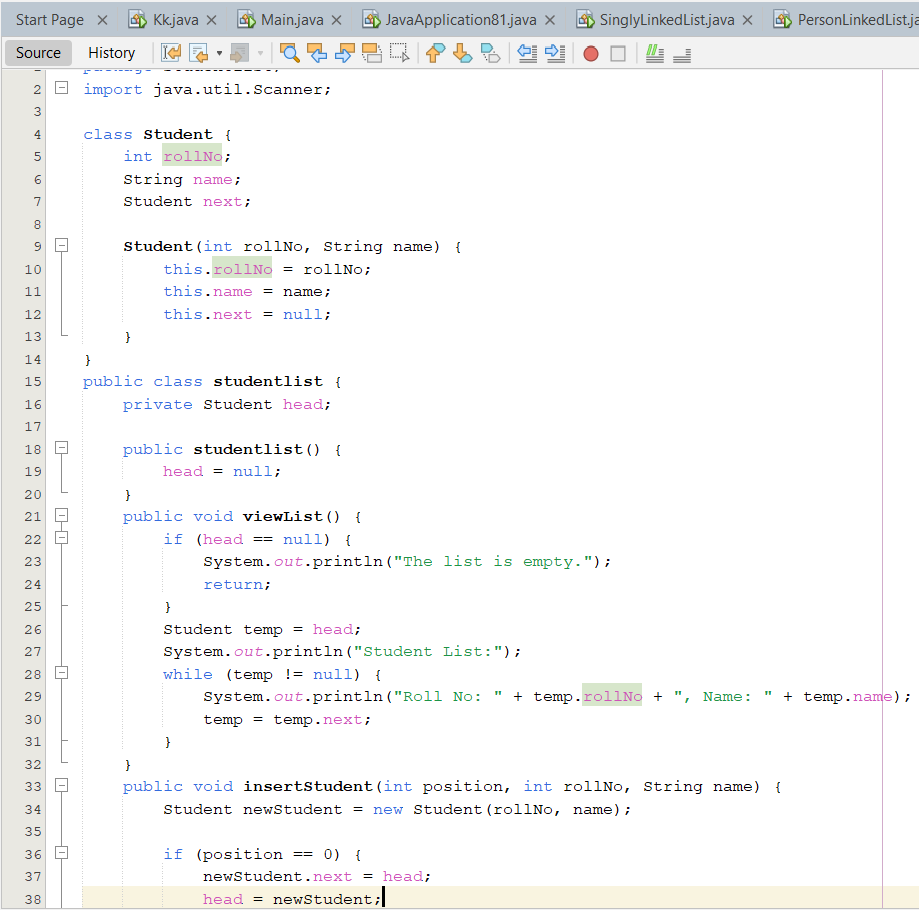
***LAB TASKS***

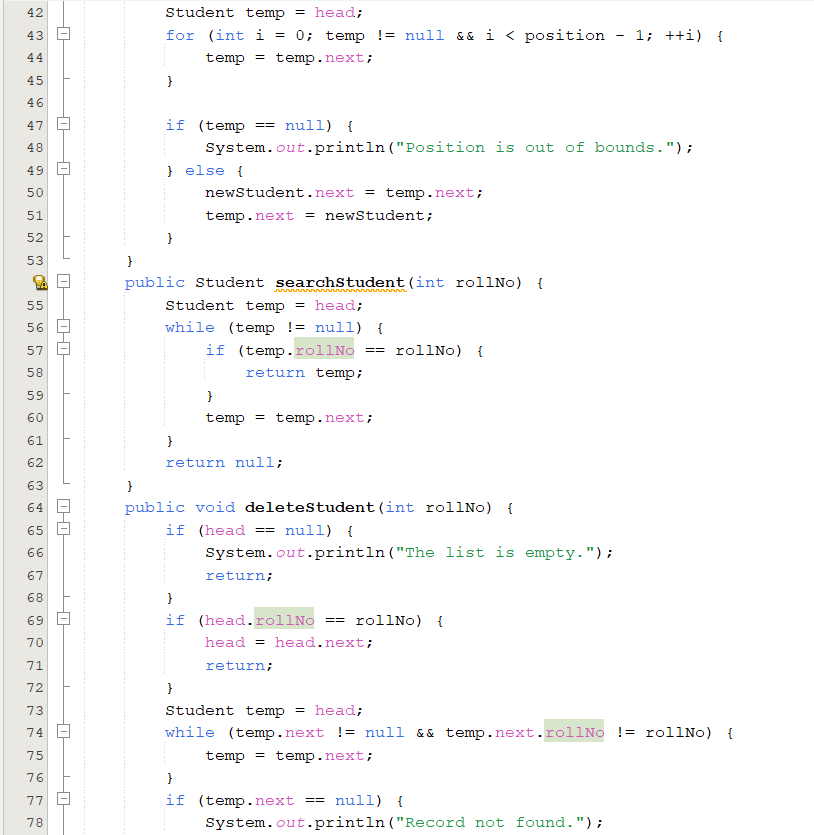
***TASK # 01***

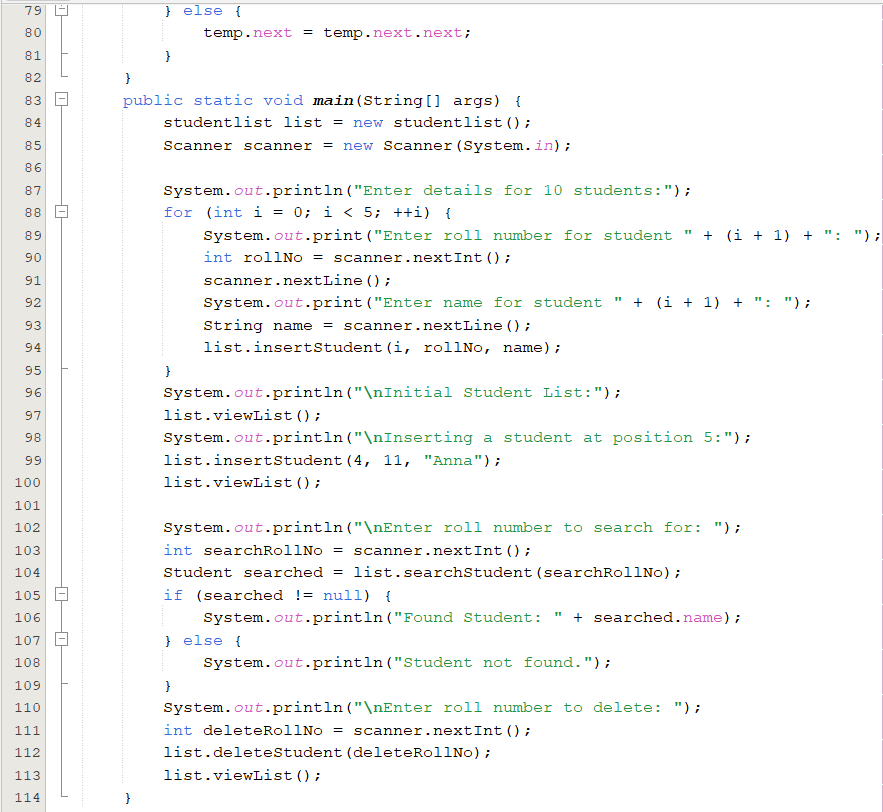
Write a program that can store 10 records of students in a link list manner and apply the following operations on it.

1. View the list
2. Insert the elements in different locations of linked list and view it.
3. Search any element from the linked list
4. Delete record again view the list after deletion.

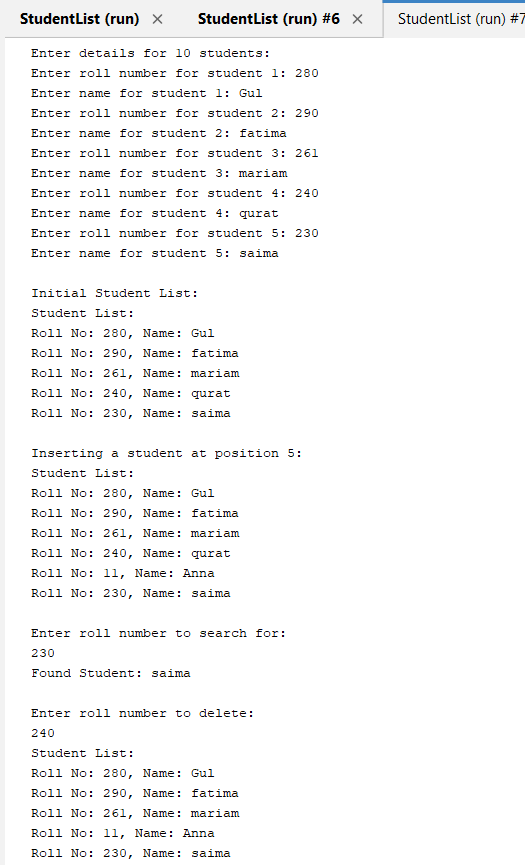
*INPUT*







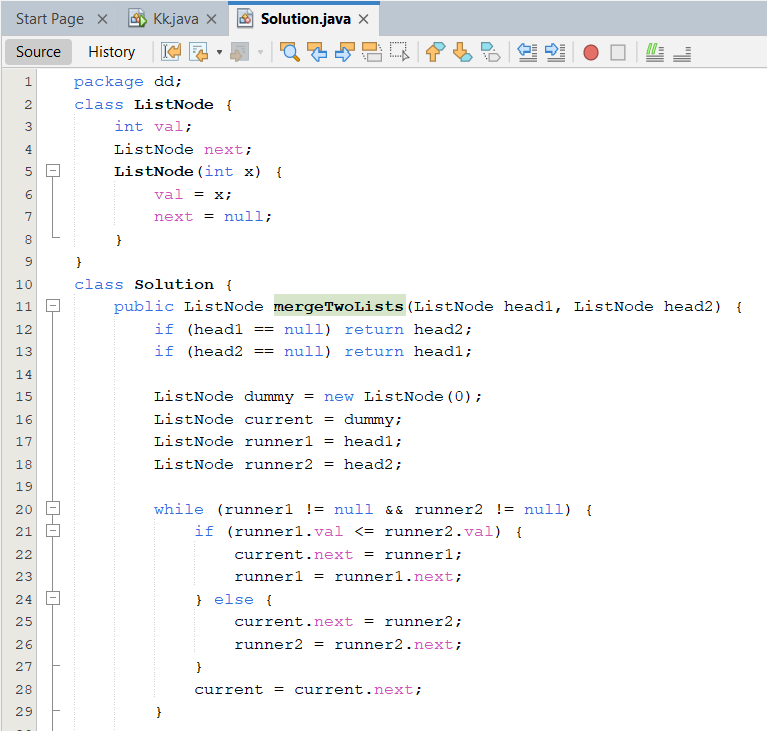
*OUTPUT*

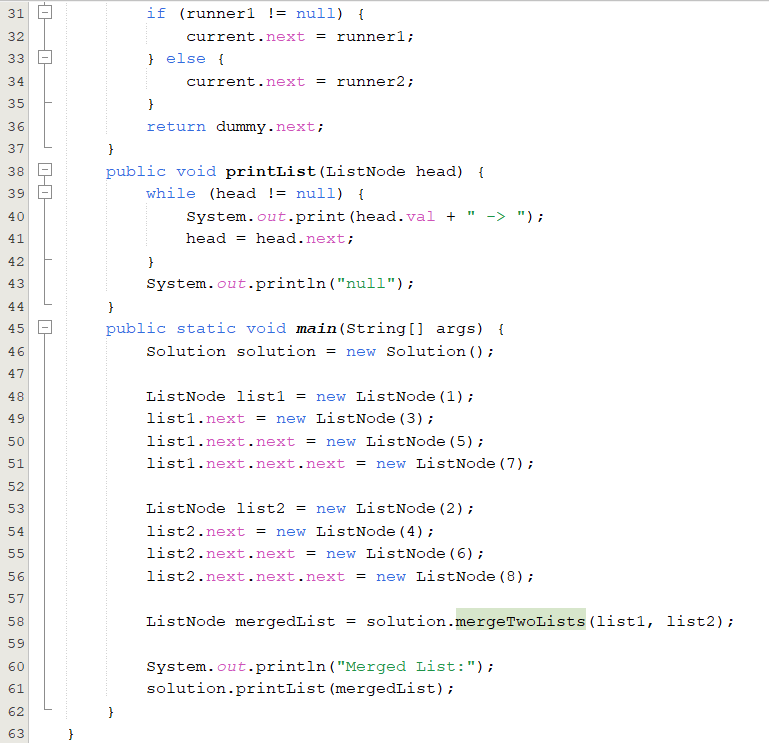


***TASK # 02***

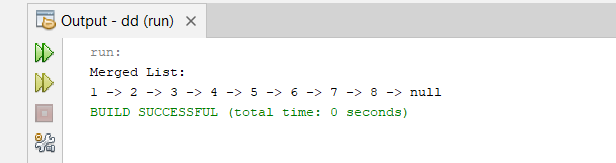
1. Write a java program to merge two equal linkedlists using runner technique.

*INPUT*





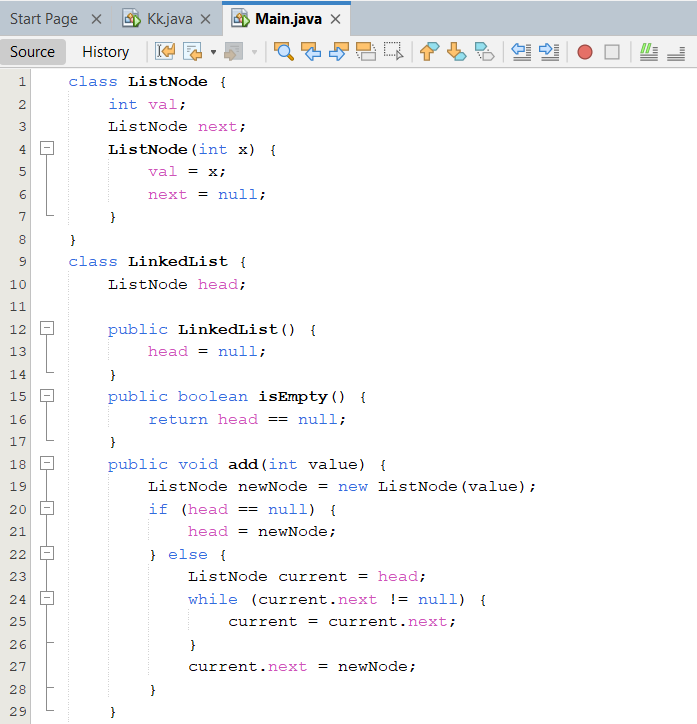
*OUTPUT*

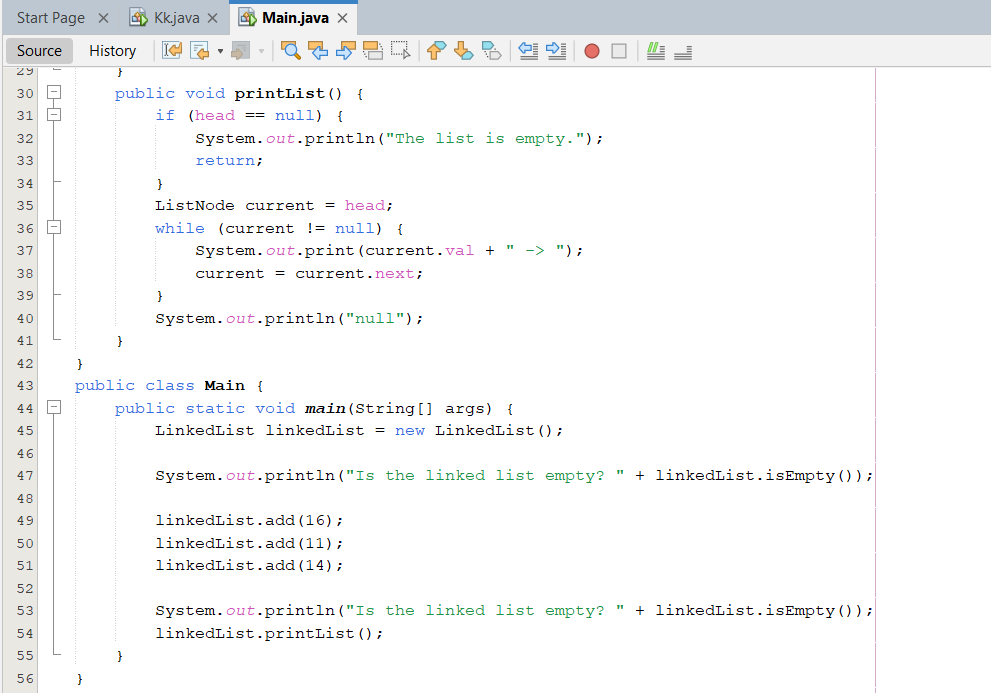


***TASK # 03***

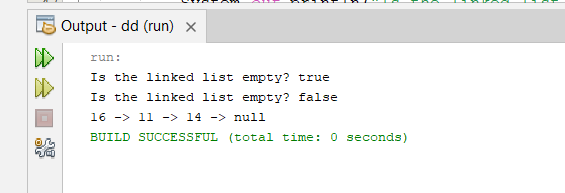
1. Write a program to check whether the linkedlist is empty or not.

*INPUT*



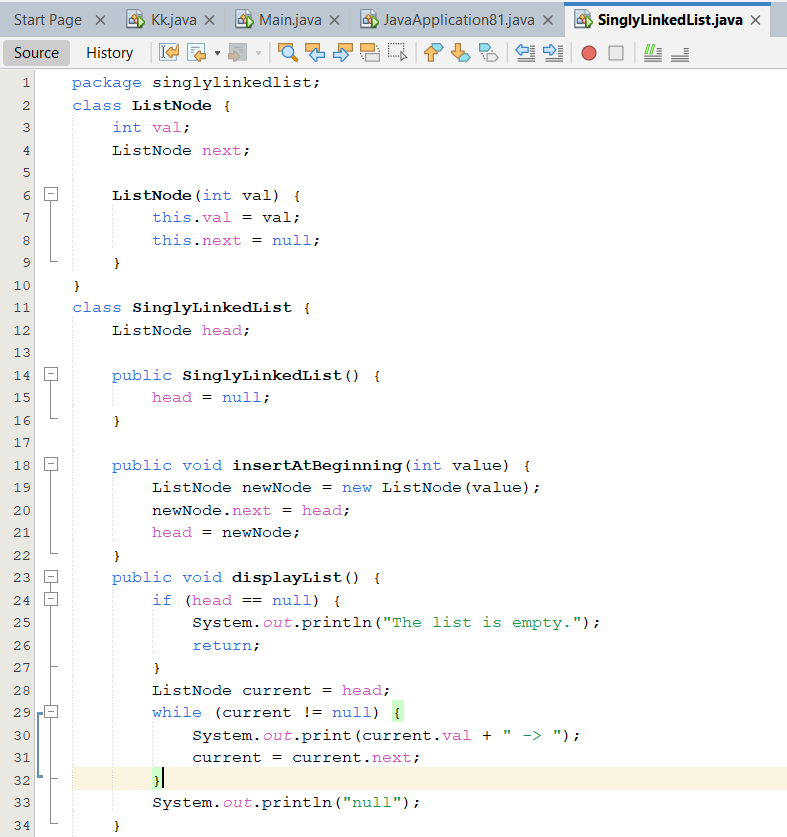


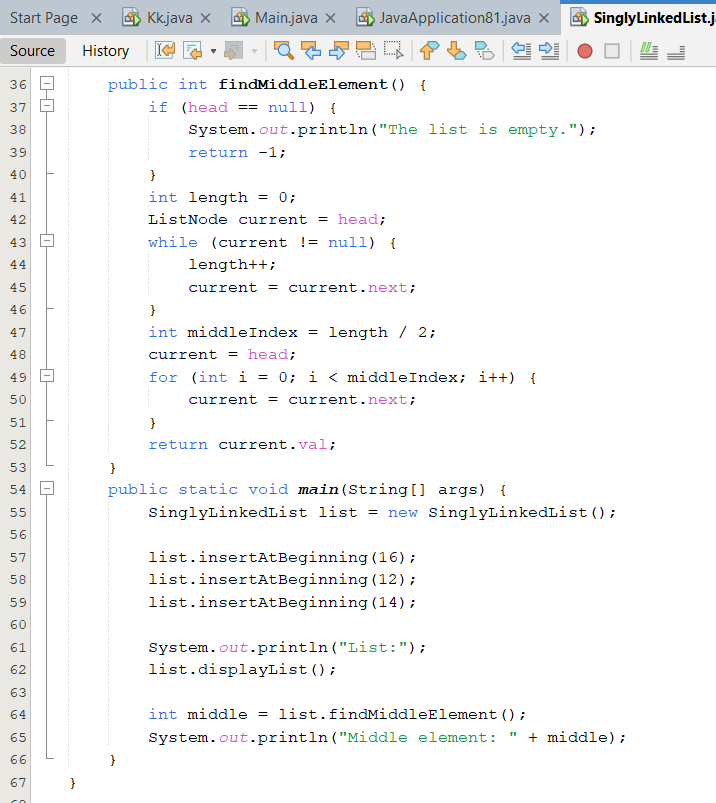
*OUTPUT*



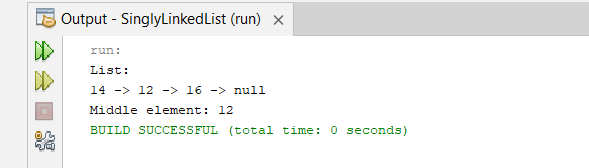
***TASK # 04***

1. You are managing a list of integers in a class, and you need to implement a **Singly Linked List** with the following operations:
2. **Insert** an integer at the **beginning** of the list.
3. **Display** the list.
4. Find the **middle element** of the list. If the list has an even number of elements, return the **first middle element**.





OUTPUT



***HOME TASKS***

***TASK # 01***

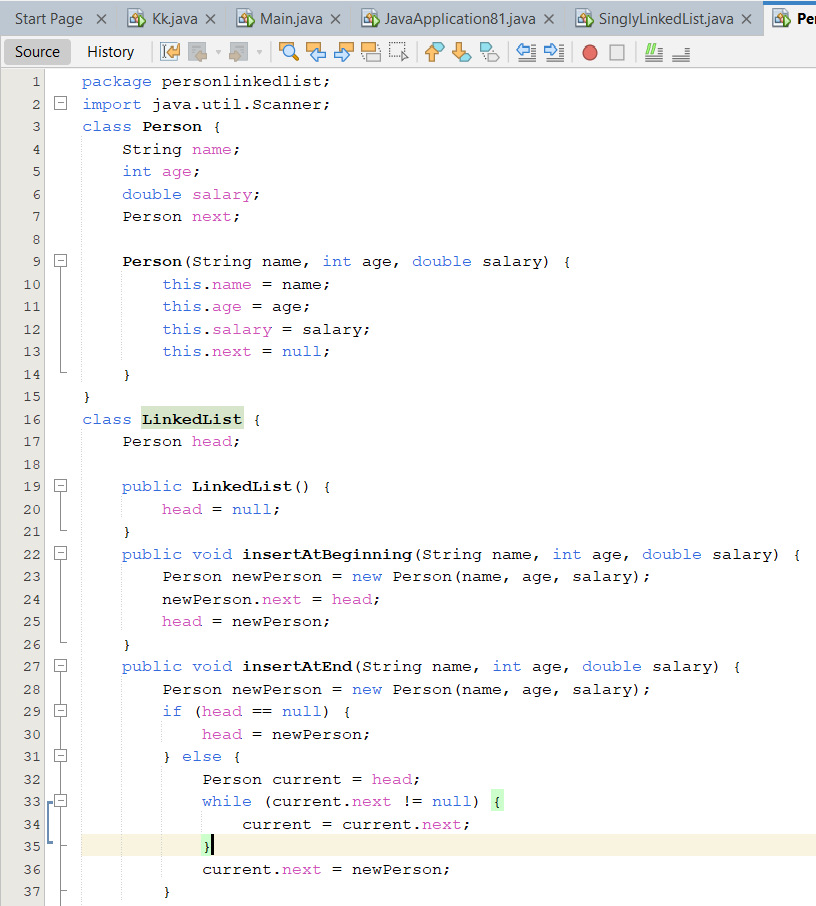
Write a program that reads the name, age and salary of 10 persons and perform the

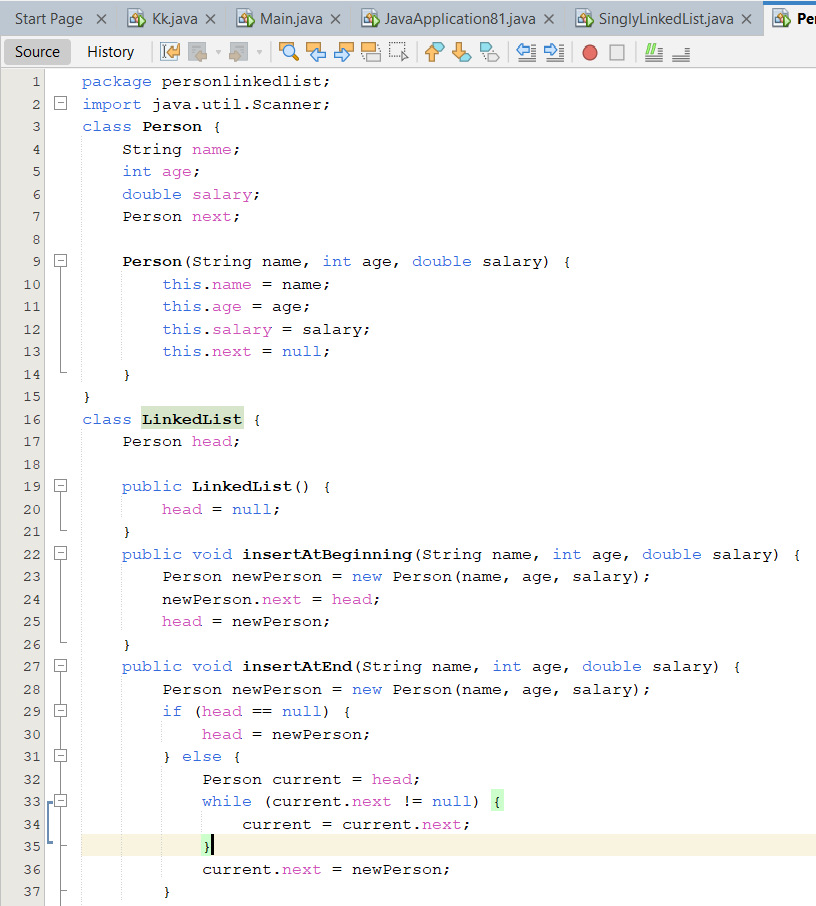
following operations on it.

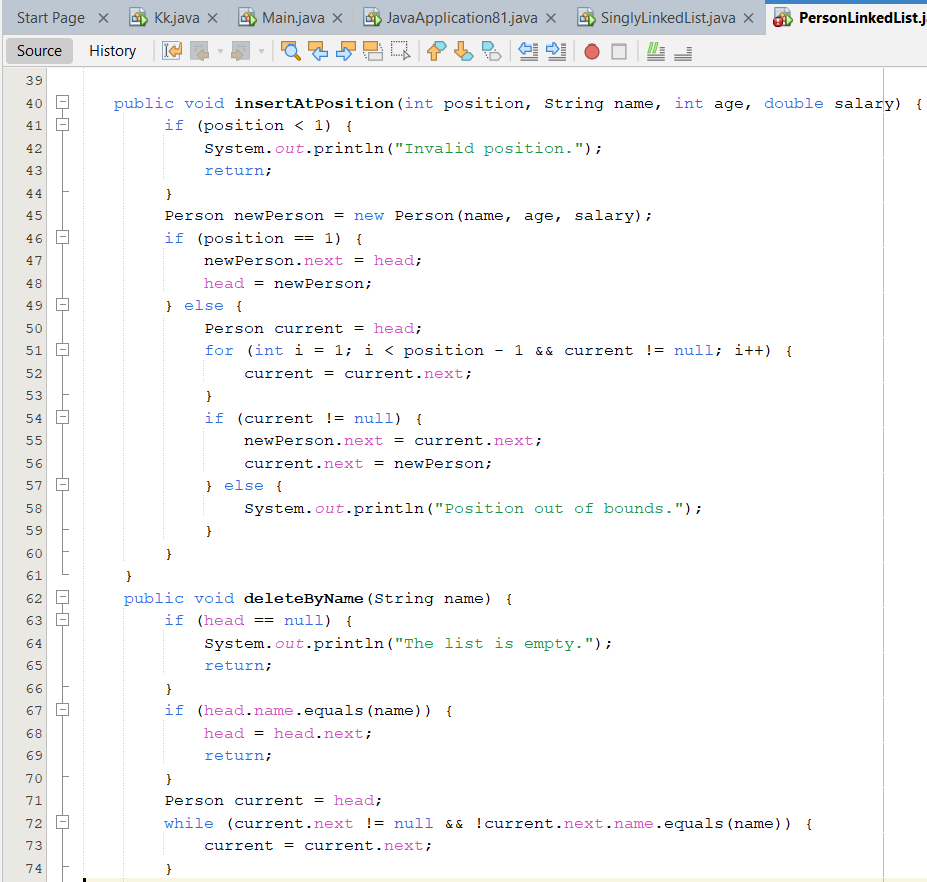
a. Insert the elements in different locations of linked list and view it.

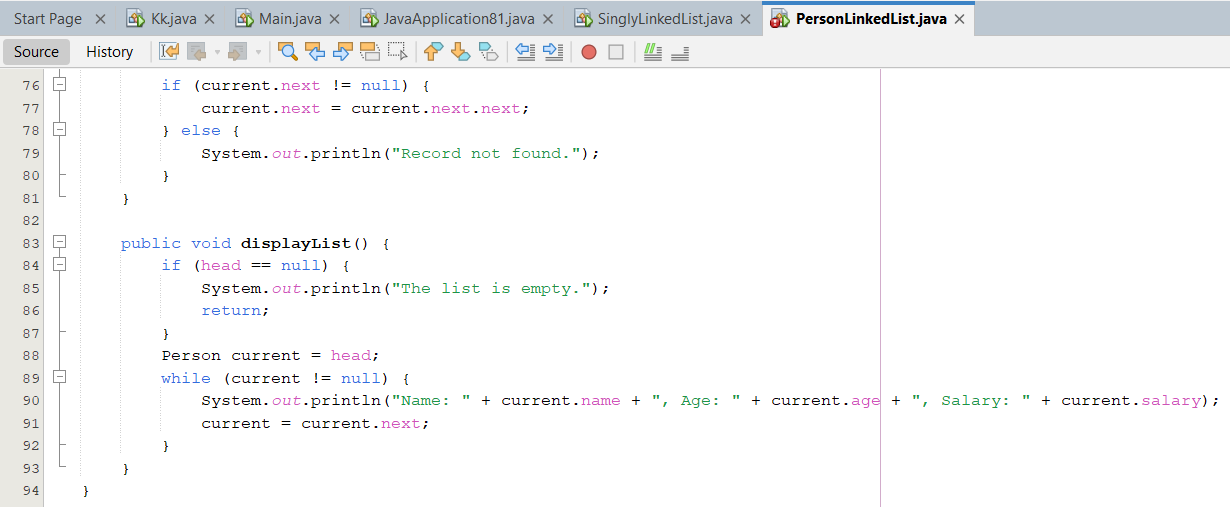
b. Delete record and again view the list after deletion

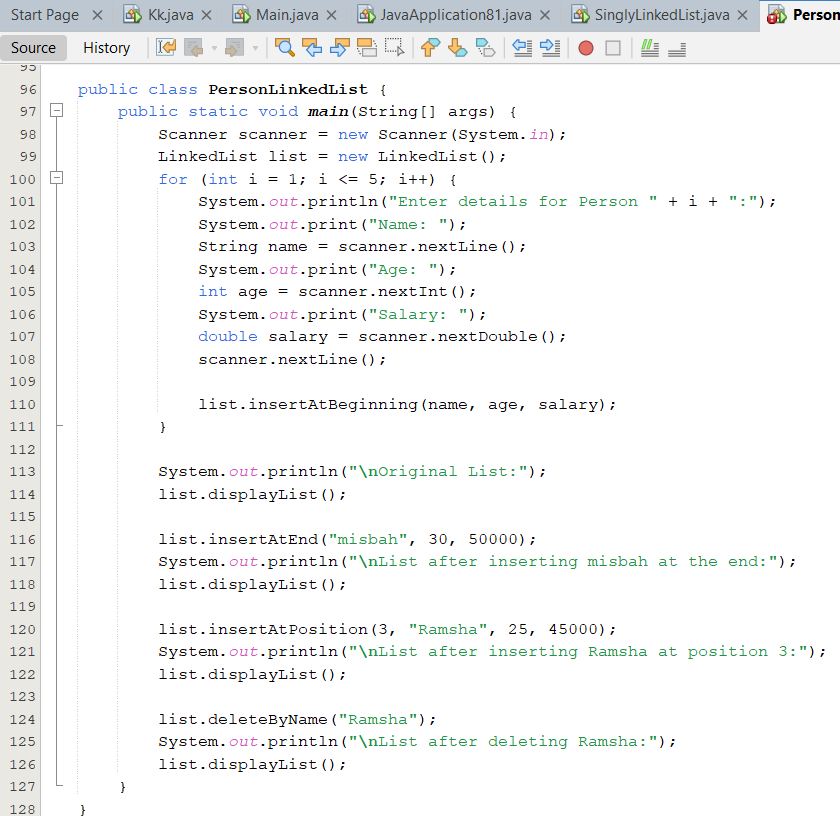
*INPUT*



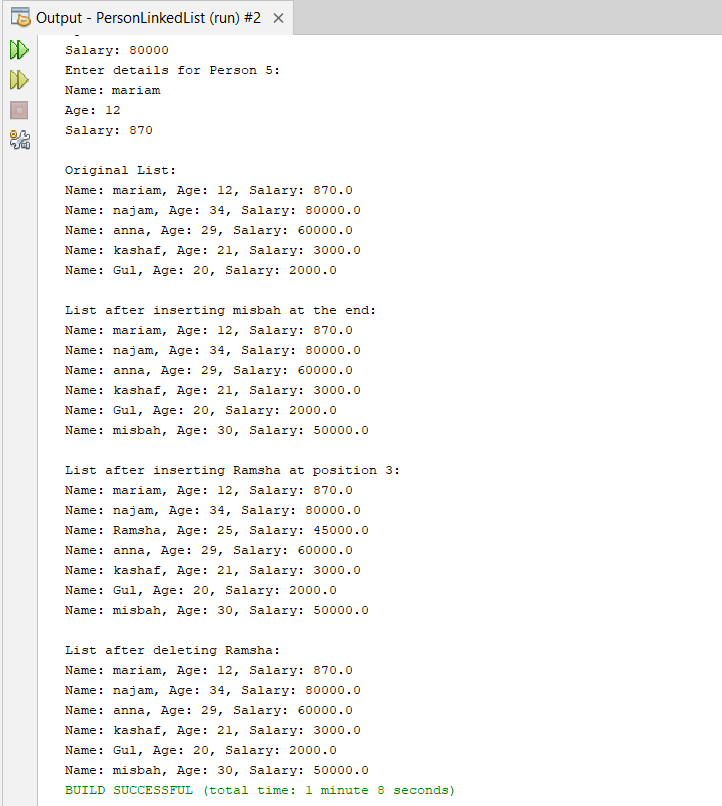








*OUTPUT*

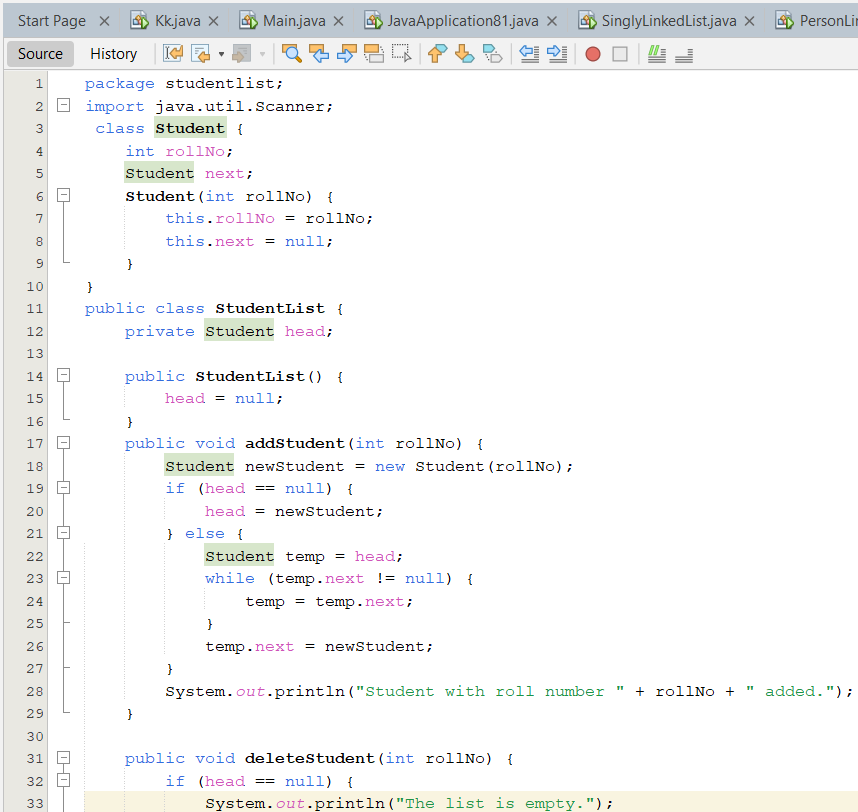


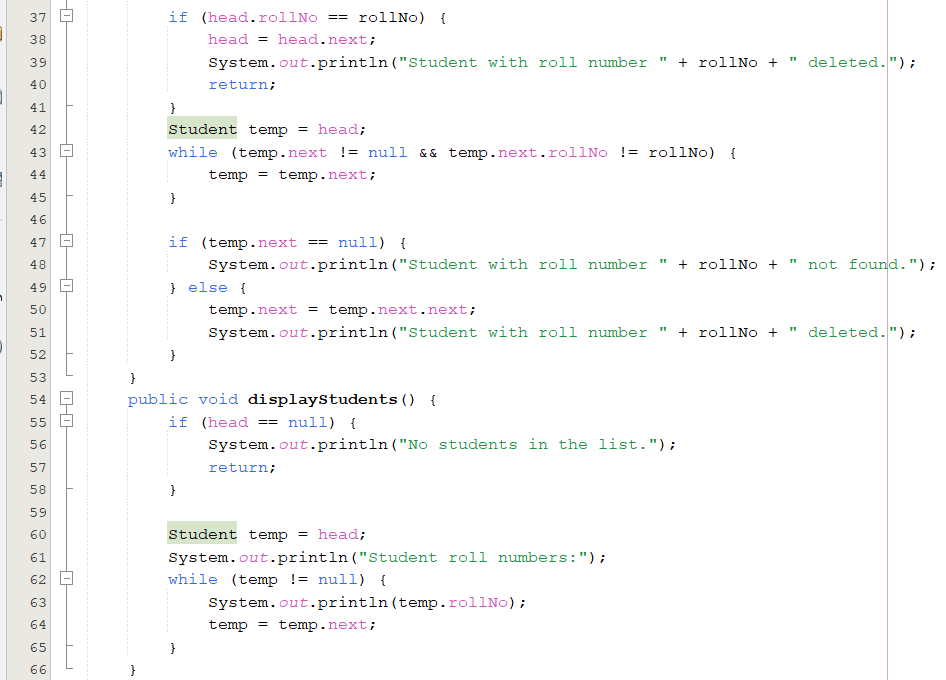
***TASK # 02***

You are tasked with managing a list of **students' roll numbers** in a class. Initially, the list is empty. You have to implement a **Singly Linked List** with the following operations:

1. **Add student roll number** at the **end** of the list.
2. **Delete a student by roll number**.
3. **Display the roll numbers** of all students in the class

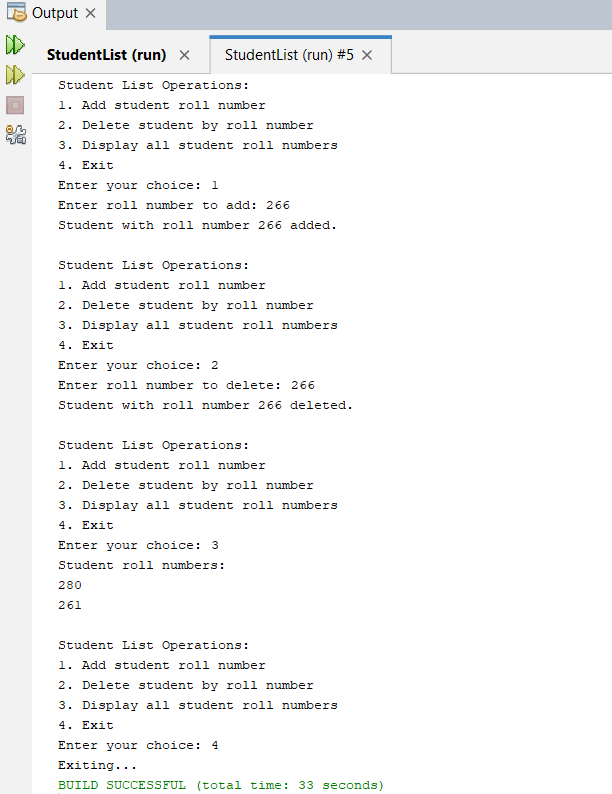
*INPUT*







*OUTPUT*

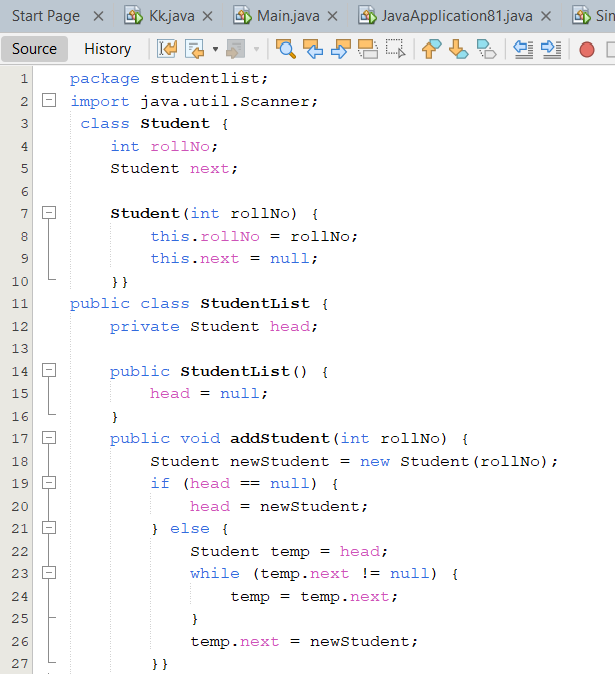


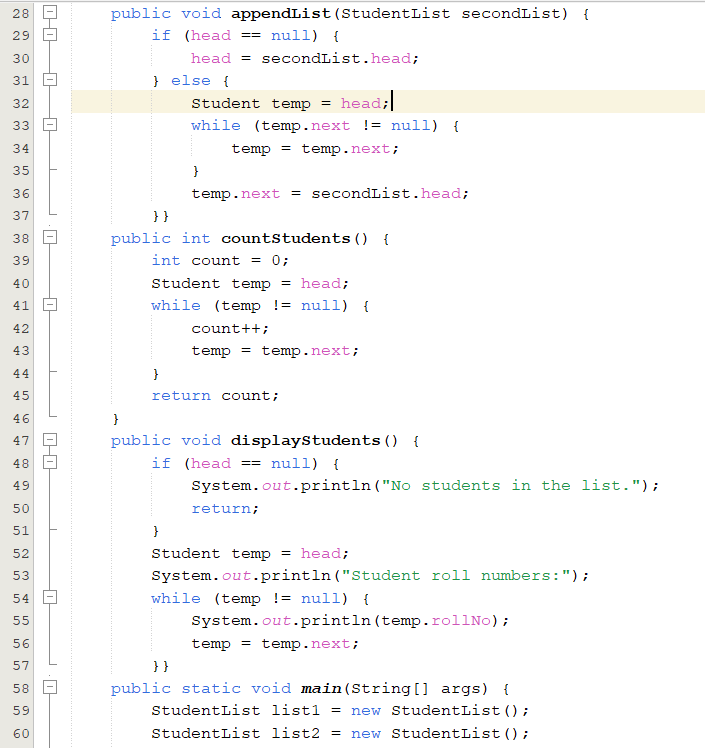
***TASK # 03***

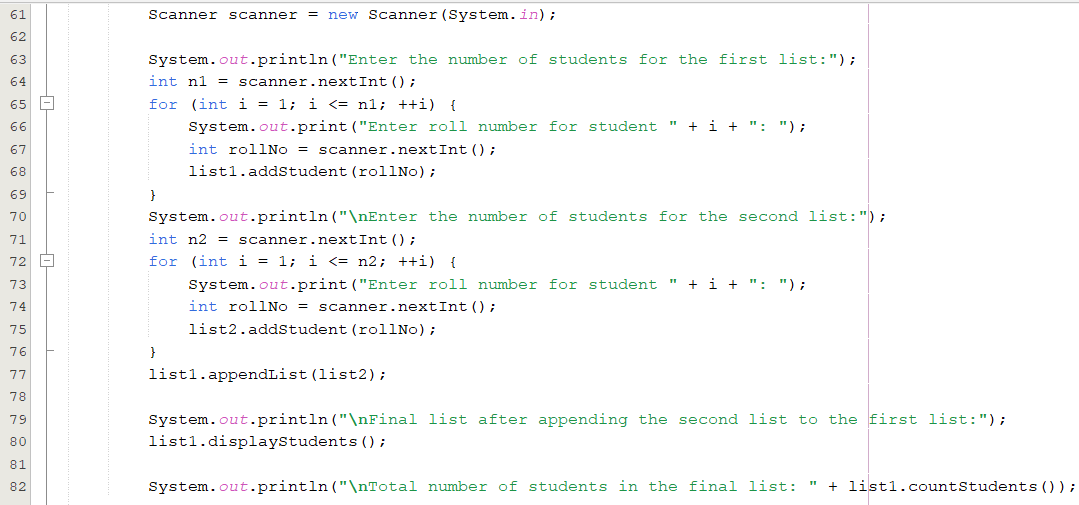
You are managing two **singly linked lists** representing **two groups of students**. Your task is to:

1. **Append** the second list to the first list (i.e., add all elements of the second list to the end of the first list).
2. **Count the number of students** in the final list (i.e., the total number of nodes in the list).
3. **Display the final list** after the append operation.

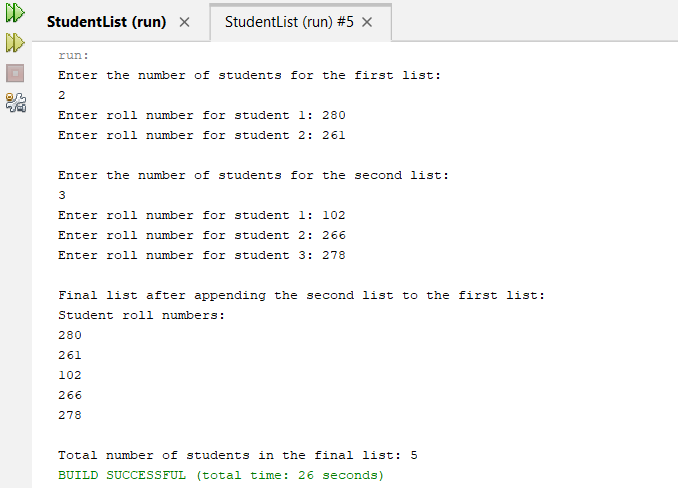
*INPUT*







*OUTPUT*



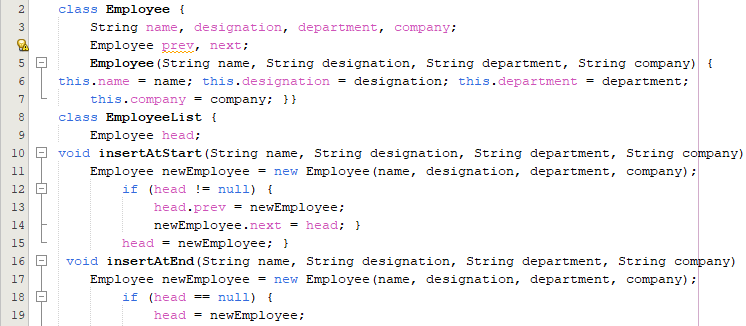
***DOUBLY LINKEDLIST***

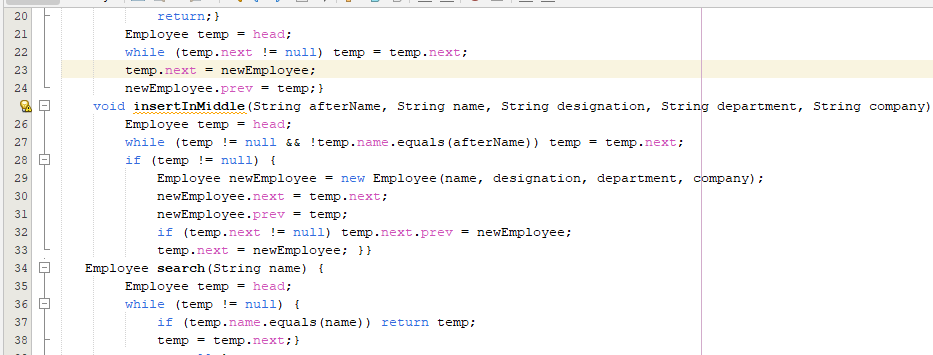
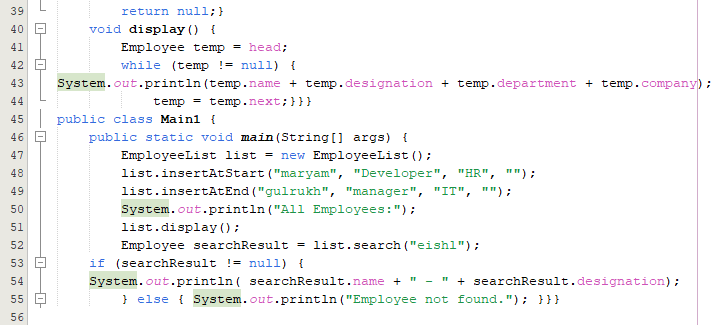
***LAB TASKS***

***TASK # 01***

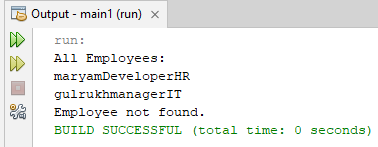
1. Write a program that can insert the records of employees in a link list. The record includes employee’s name, designation, department and company name. The program should be able to insert the record as first, last and as middle node in the list and search any record.

*INPUT*

**

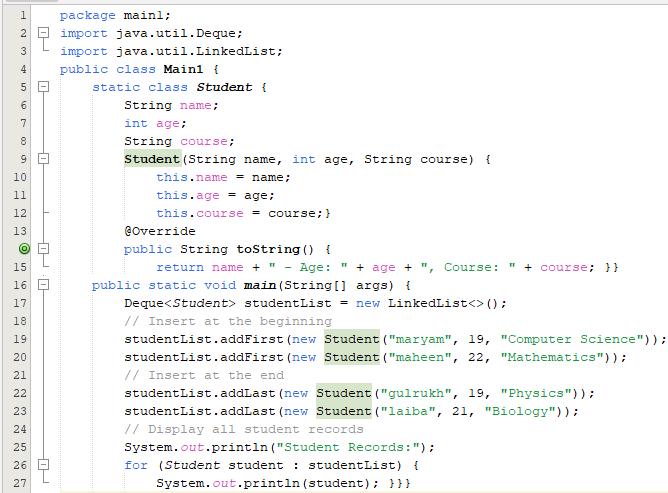
**  

*OUTPUT*

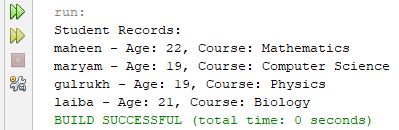
**

***TASK # 02***

Write a program to insert the records of students in a Doubly linked list and insert elements at first and last node using Deque



OUTPUT



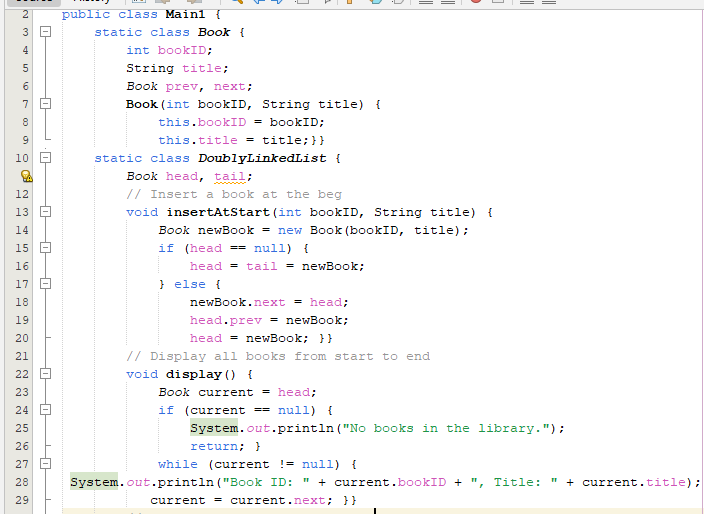
***TASK # 03***

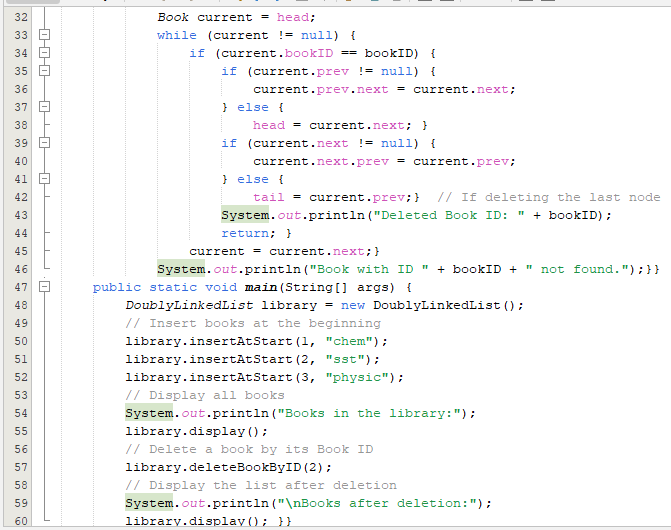
1. You are managing a **library system** where each book is represented by a node in a **doubly linked list**. Each node contains:

* **Book ID** (integer)
* **Book Title** (string)

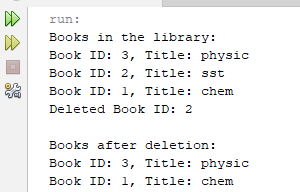
Your task is to:

1. **Insert** a book at the **beginning** of the list.
2. **Display** all books in the list from **start to end**.
3. **Delete** a book by its **Book ID**.
4. **Display** the list after deletion





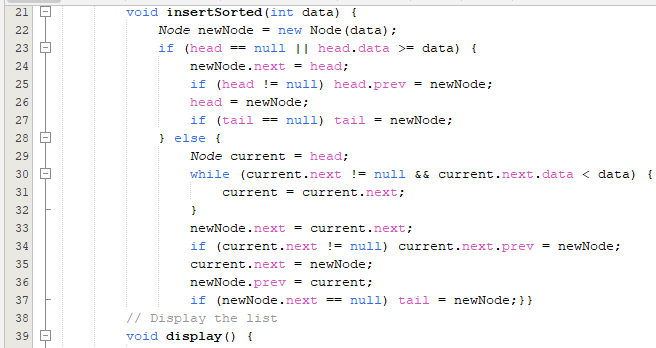
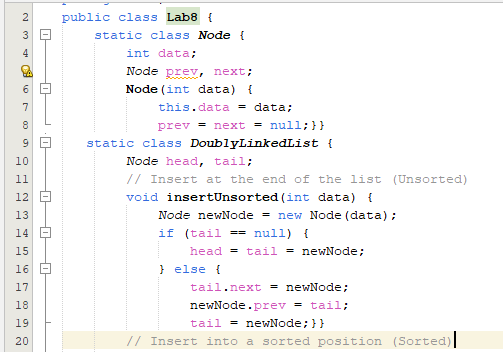
OUTPUT

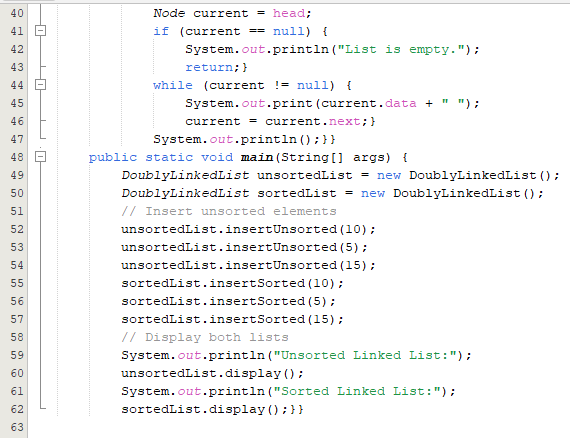


*HOME TASK*

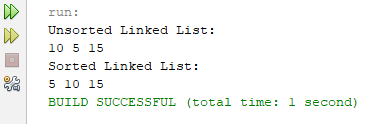
***TASK # 01***

1. Write a program to create Unsorted LinkedList as well as Sorted LinkedList.

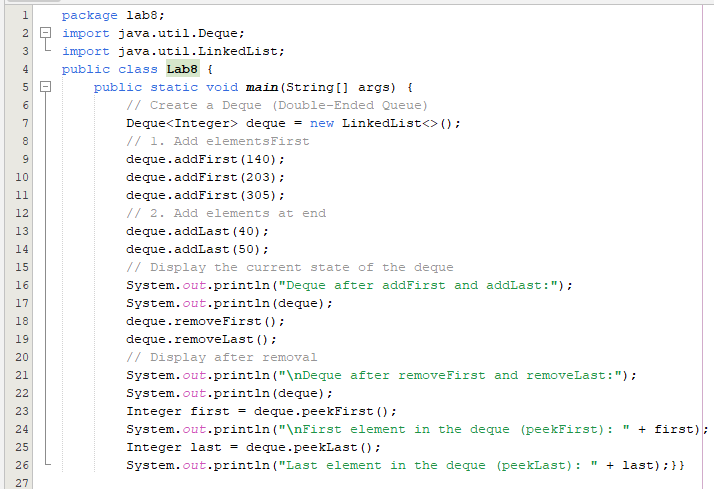


OUTPU T

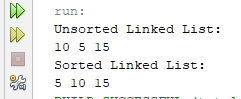


***TASK # 02***

Write a program to create LinkedList using Deque and apply any five methods of Deque interface



Output

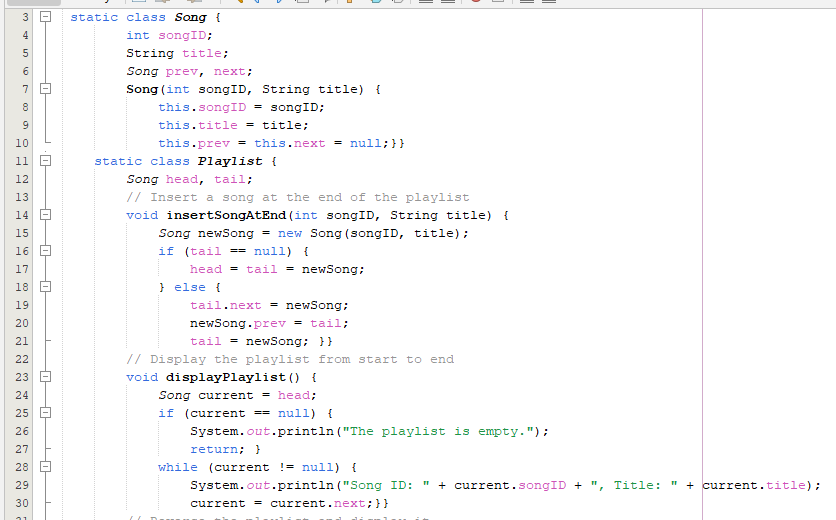


***TASK # 03***

1. You are managing a **playlist system** where each song is represented by a node in a **doubly linked list**. Each node contains:

* **Song ID** (integer)
* **Song Title** (string)

Your task is to:

1. **Insert** a song at the **end** of the playlist.
2. **Display** the playlist from **start to end**.
3. **Reverse** the playlist and display it again.
4. 

Output

