| Program Name: | Lab 07 |
| --- | --- |
| Roll No: | 1510 |
| Title of Program: | Singly Linked List |
| Objective: | code for Singly Linked List with options to:  (i) Insert at tail and at head  (ii) Search  (iii) Count  (iv) Display |

**CODE:**

**/\***

**Name: Advait Dhakad**

**Roll no: 1510**

**Unit 4: Lists**

**Program: Signly Linked List \*/**

**import java.util.Scanner;**

**// 1. Node Template**

**class Node{**

**int data;**

**Node next;**

**// Constructor**

**Node(int data, Node next){**

**this.data = data;**

**this.next = next;**

**}**

**Node(int data){**

**this.data =data;**

**this.next = null;**

**}**

**} // end of class Node**

**// 2. class list**

**class List{**

**Node head;**

**Node tail;**

**// Constructor**

**List(){**

**head = null;**

**tail = null;**

**}**

**// insert at the end of SLL**

**void InsertTail(int y){**

**Node x = new Node(y);**

**if (head==null){**

**head = x;**

**tail = x;**

**}**

**else{**

**tail.next = x; // link x to tail of the list**

**tail = x; // update tail to x**

**}**

**}// end of insertTail**

**// count the number of node**

**void Count(){**

**if (head==null){**

**System.out.println("Empty List!!! 0 elements");**

**}**

**else{**

**int count = 0;**

**Node tmp = head;**

**while(tmp != null){**

**count++;**

**tmp = tmp.next;}**

**System.out.println("The count of elements is: "+ count);**

**}// end of count**

**}**

**// display the SLL**

**void Display(){**

**if (head==null){**

**System.out.println("Empty List!!! ");**

**}**

**else{**

**Node tmp = head;**

**while(tmp != null){**

**System.out.print(" " + tmp.data+ "->");**

**tmp = tmp.next;}**

**System.out.println("\nEnd of list");}**

**}// end of Display**

**// search for a node**

**void Search(int x){**

**if (head==null){**

**System.out.println("Empty List!!!");**

**}**

**else{**

**int flg = 0;**

**Node tmp = head;**

**while(tmp != null){**

**if (tmp.data == x){**

**flg=1;**

**break;}**

**tmp = tmp.next;}**

**if(flg==1)**

**System.out.println("Element found!!");**

**else**

**System.out.println("Element not found!!");**

**}**

**}// end of Search**

**// delete the node**

**} // end of class List**

**// 3. Interface**

**class SLL{**

**public static void main(String[] args){**

**Scanner scan = new Scanner(System.in);**

**// Create object of lists**

**List s = new List();**

**char ch;**

**do{**

**System.out.println("\n");**

**System.out.println("\t\*\*\*\*\*\*\* SINGLE LINKED LIST \*\*\*\*\*\*\*\*\*\*\*\*\* \n");**

**System.out.println(" 1 . Insert at the end of the SLL");**

**System.out.println(" 2 . Count the number of nodes");**

**System.out.println(" 3 . Display the SLL");**

**System.out.println(" 4 . Search for a Node");**

**System.out.println(" 5 . Delete a node \n");**

**System.out.print("Enter your Choice: ");**

**int choice = scan.nextInt();**

**switch(choice){**

**case 1:**

**System.out.println("Selected to insert the data");**

**System.out.print("Enter the data you want insert: ");**

**s.InsertTail(scan.nextInt());**

**break;**

**case 2:**

**System.out.println("Selected to Count the number of nodes");**

**s.Count();**

**break;**

**case 3:**

**System.out.println("Selected to Display the SLL");**

**s.Display();**

**break;**

**case 4:**

**System.out.println("Selected to Search for a node");**

**System.out.print("Enter the data you want Search:");**

**s.Search(scan.nextInt());**

**break;**

**case 5:**

**System.out.println("Selected to Delete a node");**

**break;**

**default:**

**System.out.println("Wrong choice!!");**

**break;**

**}// end of switch**

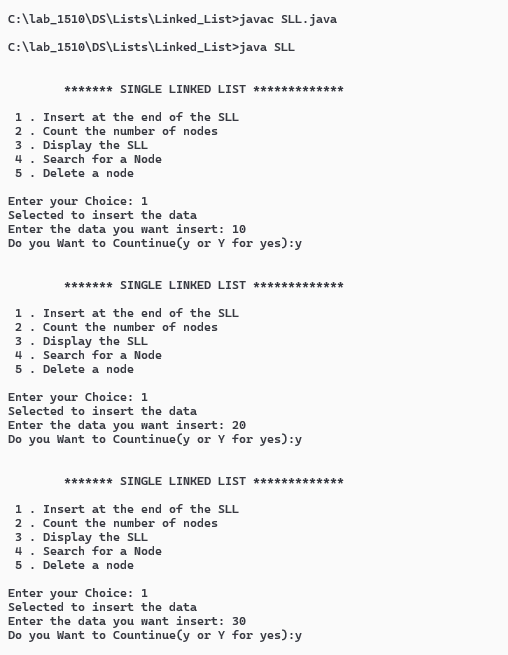
**System.out.print("Do you Want to Countinue(y or Y for yes):" );**

**ch = scan.next().charAt(0);**

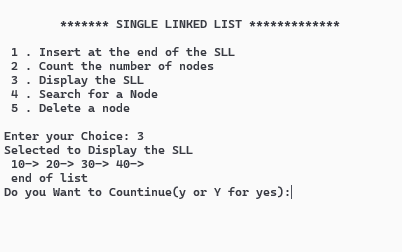
**}while(ch=='y' || ch=='Y'); // end of while } // end of main} // end of class SLL**

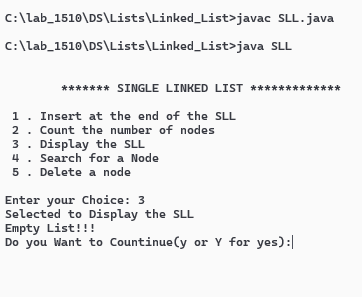
**OUTPUT:**

**Entering** the elements:

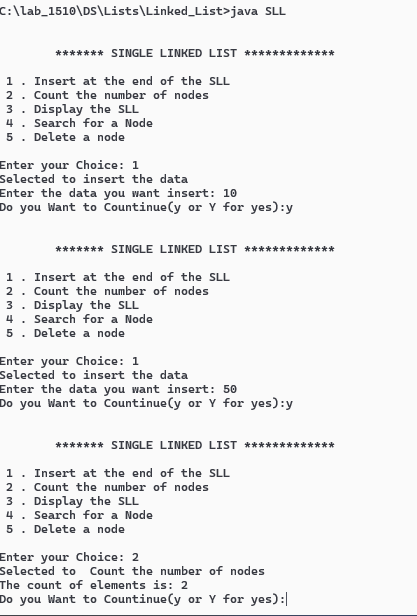
****

**Displaying** the list:

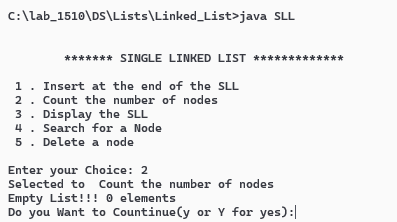
****

****

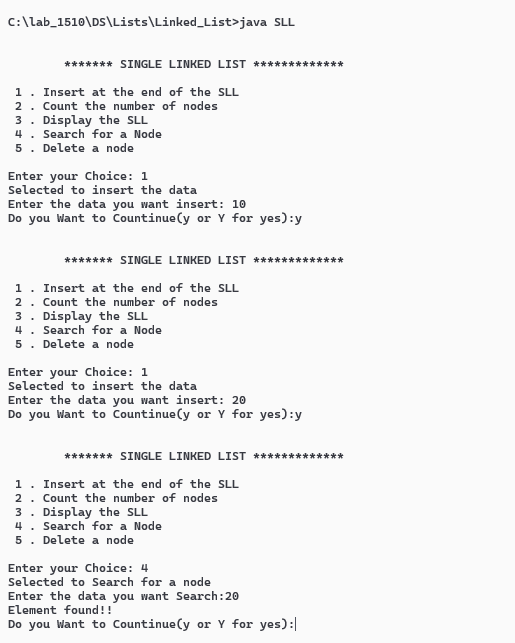
**Counting** the number of elements:

****

When the list is empty

****

**Searching** for a node:



Element not found

