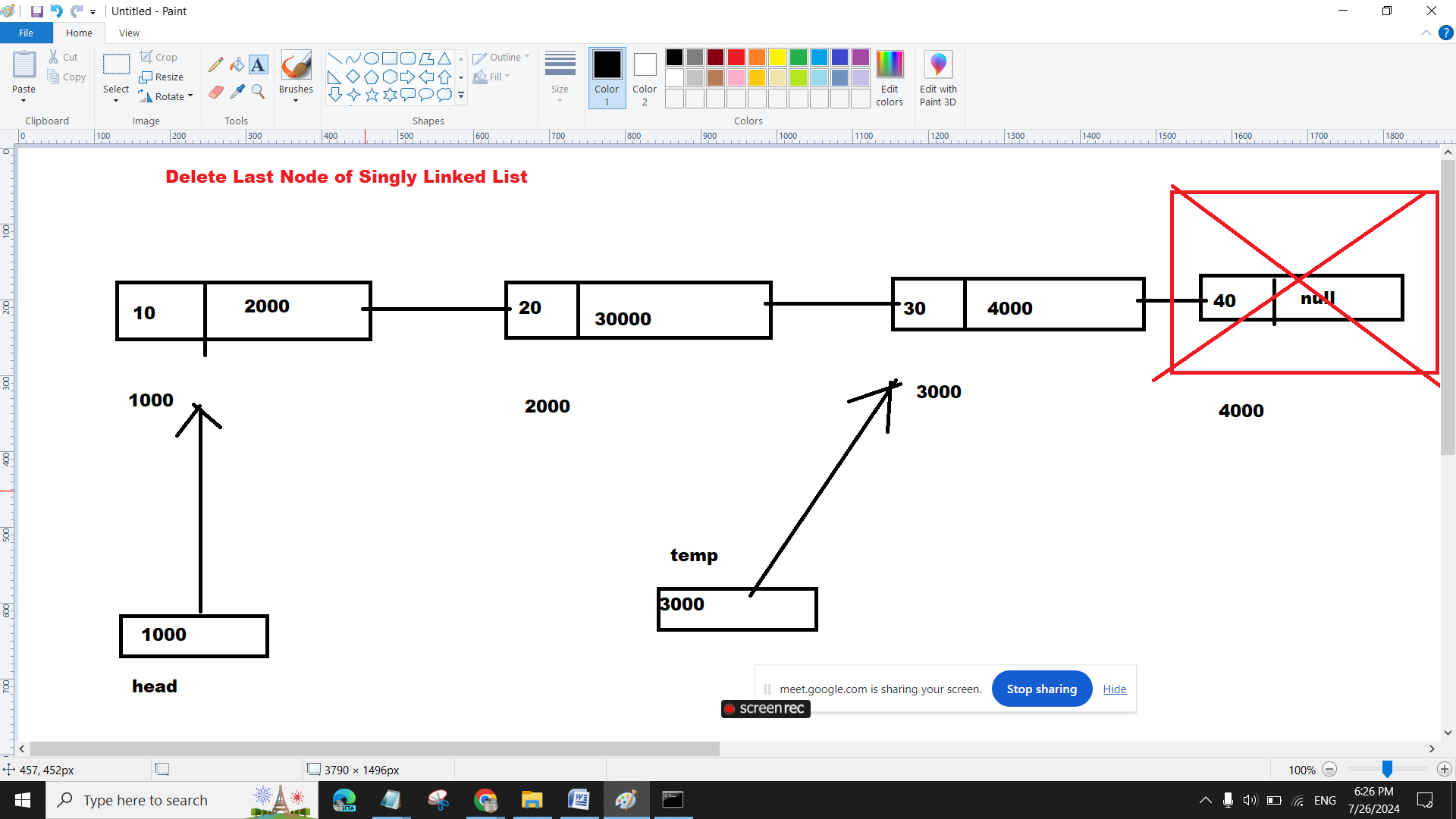
**Q1. Write a java program to delete last Node of Singly Linked List?**



class Node {

int data;

Node next;

// Constructor

public Node(int data) {

this.data = data;

this.next = null;

}

// Method to add a node at the start of the list

public Node addAtStart(Node head, int d) {

Node newNode = new Node(d);

newNode.next = head;

head = newNode;

return head;

}

// Method to add a node at the end of the list

public Node addAtEnd(Node head, int d) {

Node newNode = new Node(d);

if (head == null) {

head = newNode;

} else {

Node temp = head;

while (temp.next != null) {

temp = temp.next;

}

temp.next = newNode;

}

return head;

}

// Method to print the data in the list

public void printData(Node head) {

if (head == null) {

System.out.println("List is Empty");

} else {

Node temp = head;

while (temp != null) {

System.out.print("==>" + temp.data);

temp = temp.next;

}

}

}

public Node deleteLast(Node head){

if(head==null){

System.out.println("List is Empty");

}

if(head.next==null){

head=null;

}

Node temp=head;

while(temp.next.next!=null){

temp=temp.next;

}

temp.next=null;

return head;

}

public static void main(String[] args) {

// Create nodes

Node first = new Node(10);

Node second = new Node(20);

Node third = new Node(30);

Node fourth = new Node(40);

// Link nodes to form a list

Node head = first;

head.next = second;

head.next.next = third;

head.next.next.next = fourth;

// Print the original list

System.out.println("Print Data of Singly Linked List:");

head.printData(head);

head=head.deleteLast(head);

System.out.println("\nPrint Data of Singly Linked List after delete Last Node:");

head.printData(head);

}

}