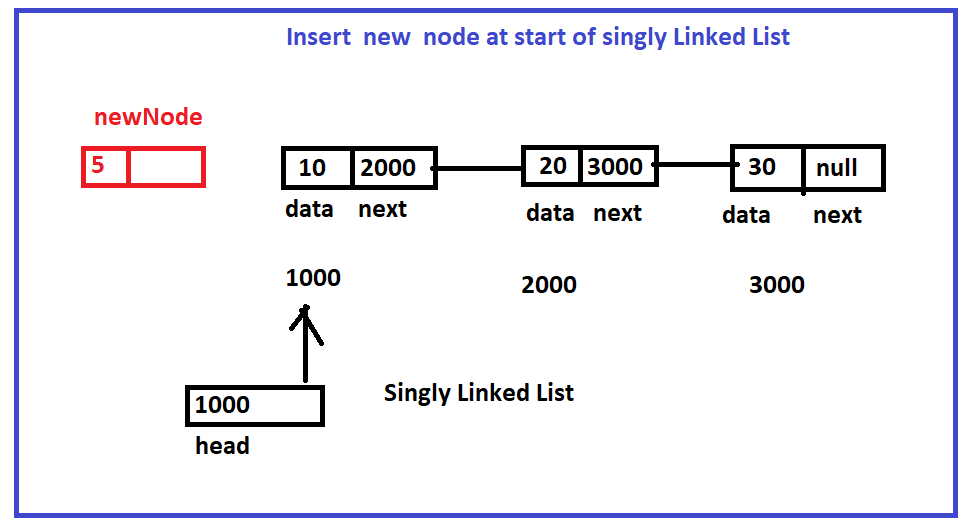
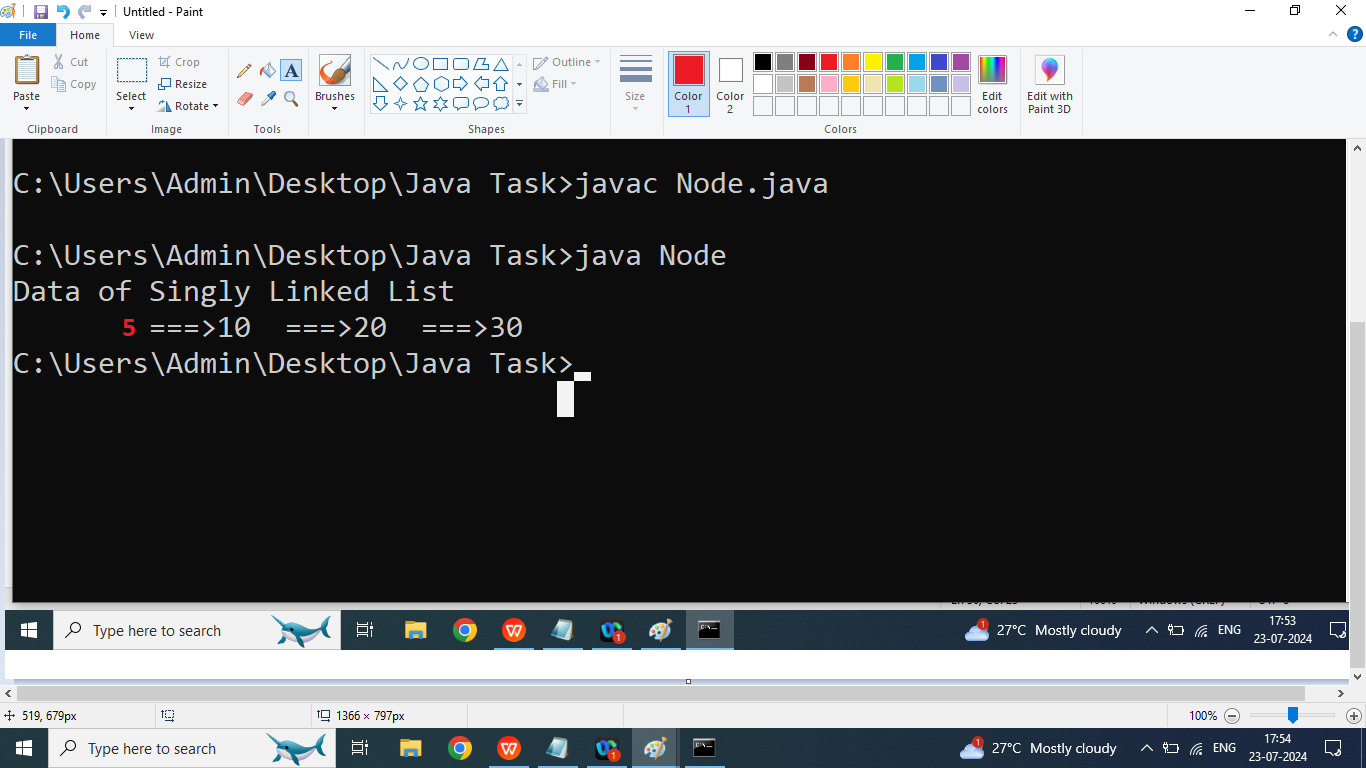
Q1. Write a java program to insert new node at starting of the singly Linked List?





class Node{

int data;

Node next;

public Node(int data){

this.data=data;

this.next=null;

}

public void printData(Node head){

if(head==null){

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

}

else{

Node temp=head;

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

while(temp!=null){

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

temp=temp.next;

}

}

}

public static void main(String args[]){

Node first=new Node(10);

Node second=new Node(20);

Node third=new Node(30);

Node head=first;

head.next=second;

head.next.next=third;

head.printData(head);

head=head.addAtStart(head,5);

System.out.println("\nData After insert new node at start");

head.printData(head);

}

public Node addAtStart(Node head,int d){

Node newNode=new Node(d);

newNode.next=head;

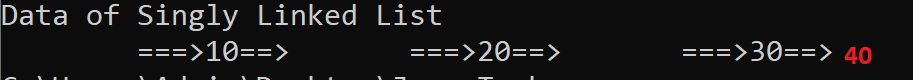
head=newNode;

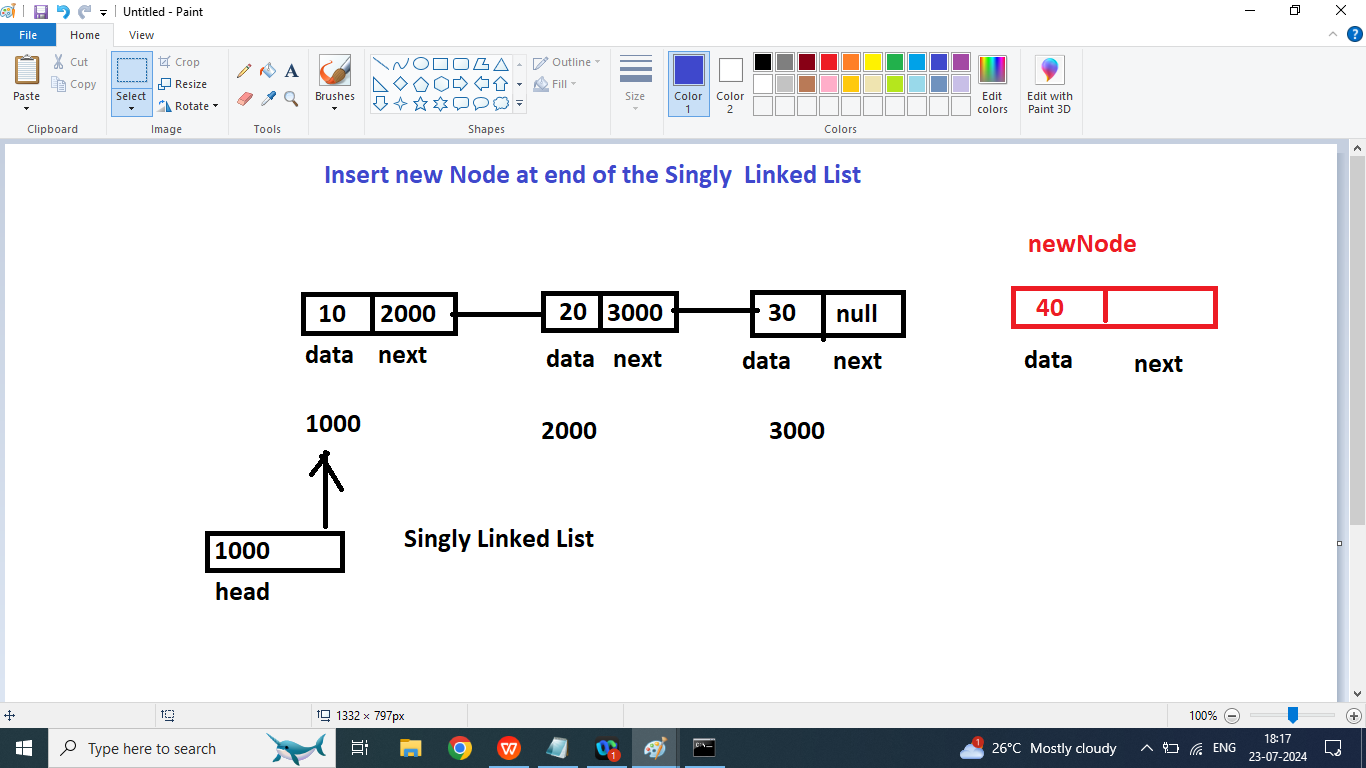
return head;

}

}

Q2. Write a java program to insert new Node at Ending of Singly Linked List?





class Node{

int data;

Node next;

public Node(int data){

this.data=data;

this.next=null;

}

public void printData(Node head){

if(head==null){

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

}

else{

Node temp=head;

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

while(temp!=null){

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

temp=temp.next;

}

}

}

public Node addAtStart(Node head,int d){

Node newNode=new Node(d);

newNode.next=head;

head=newNode;

return head;

}

public Node addAtEnd(Node head,int d){

Node newNode=new Node(d);

if(head==null){

head=newNode;

return head;

}else{

Node temp=head;

while(temp.next!=null){

temp=temp.next;

}

temp.next=newNode;

}

return head;

}

public static void main(String args[]){

Node first=new Node(10);

Node second=new Node(20);

Node third=new Node(30);

Node head=first;

head.next=second;

head.next.next=third;

head.printData(head);

System.out.println("\nPrint Data after add new Node at end");

head=head.addAtEnd(head,40);

head.printData(head);

}

public Node addAtEnd(Node head,int d){

Node newNode=new Node(d);

if(head==null){

head=newNode;

return head;

}else{

Node temp=head;

while(temp.next!=null){

temp=temp.next;

}

temp.next=newNode;

}

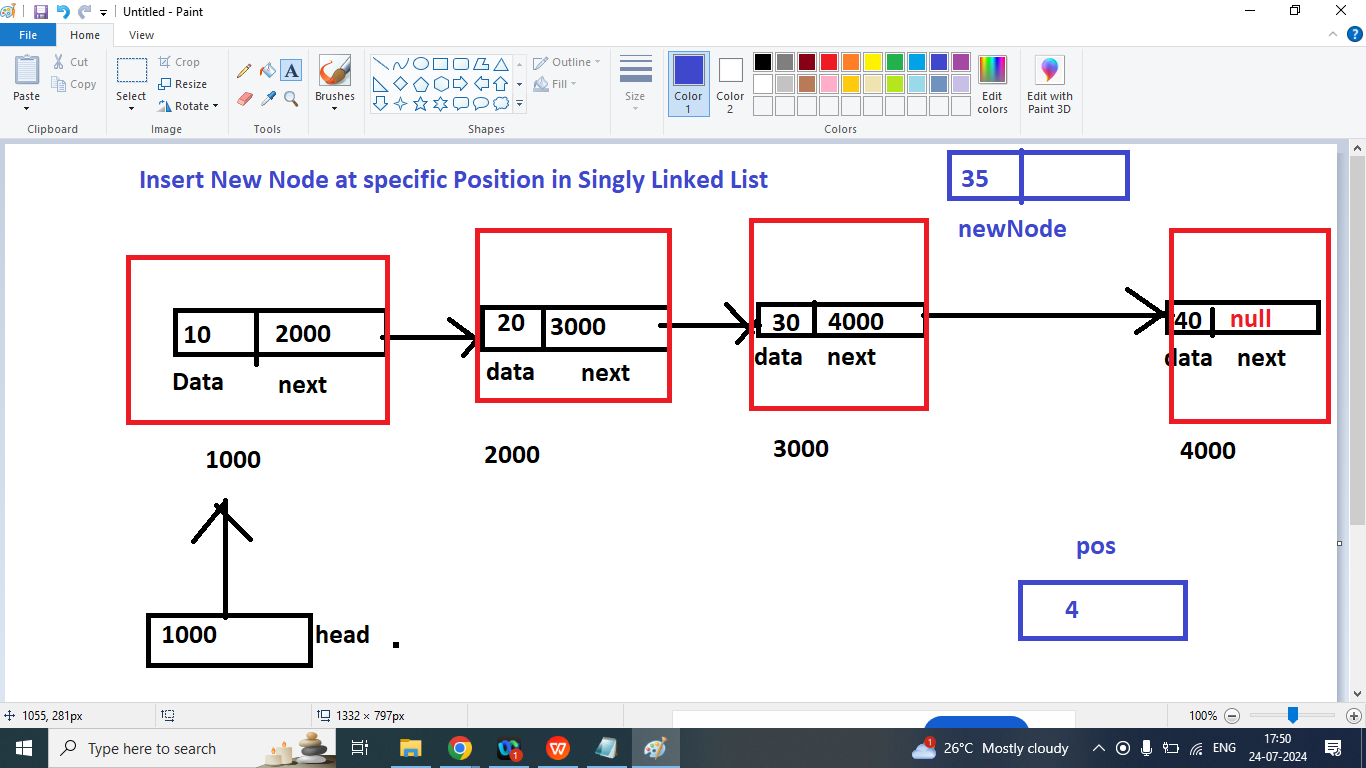
return head;

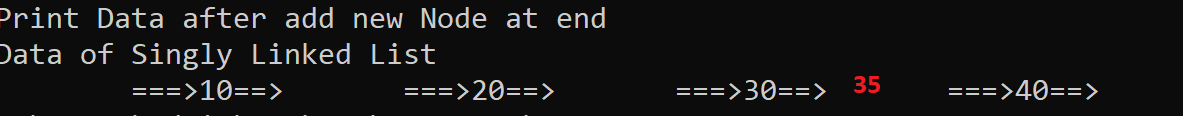
}

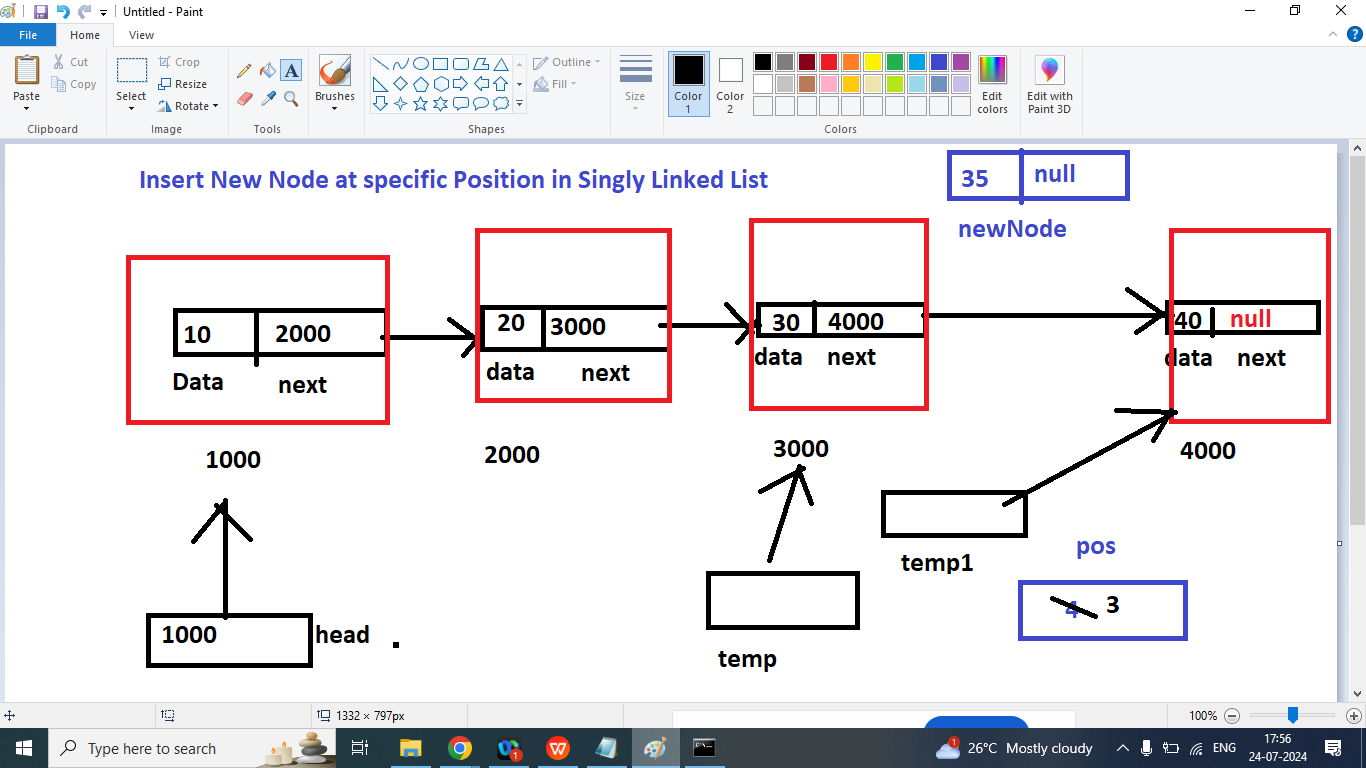
}

Q1. Write a java program to insert new node at nth Position in Singly Linked List?

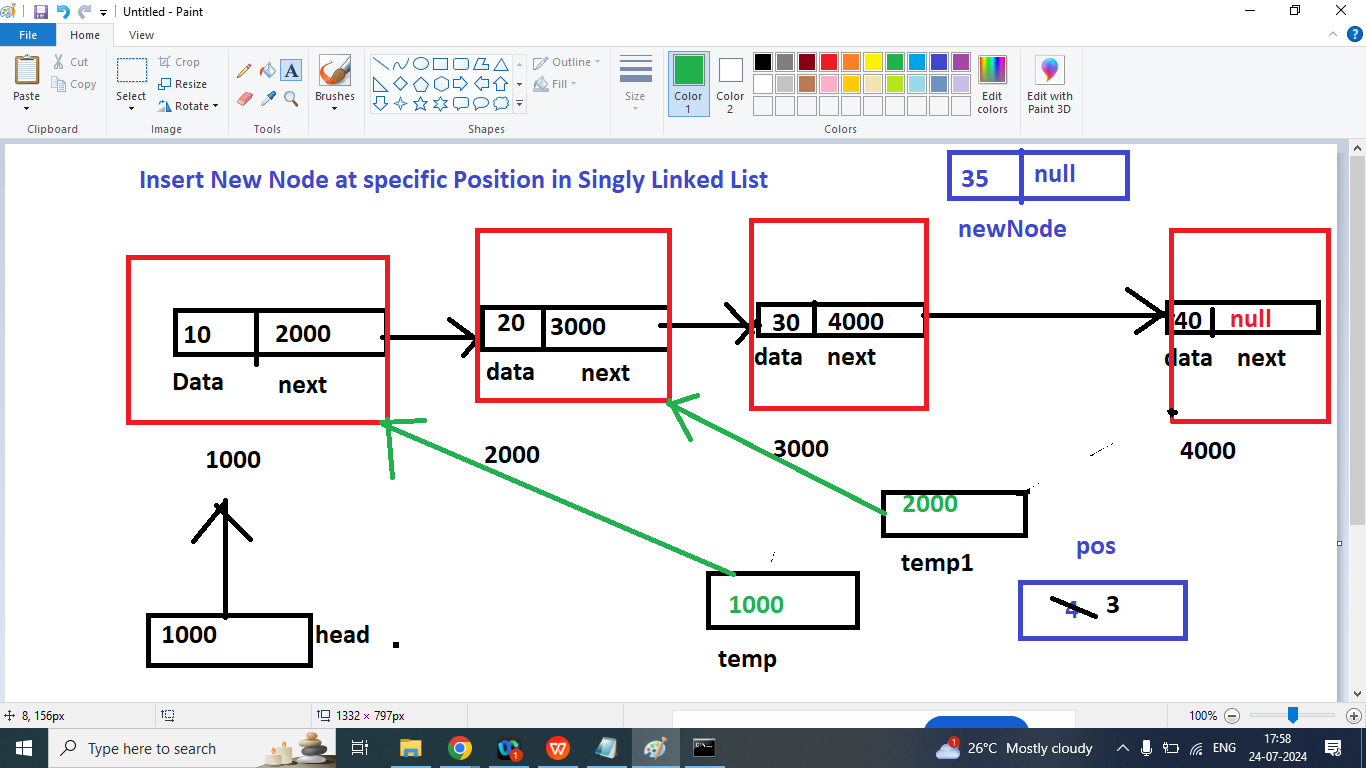
Ans:



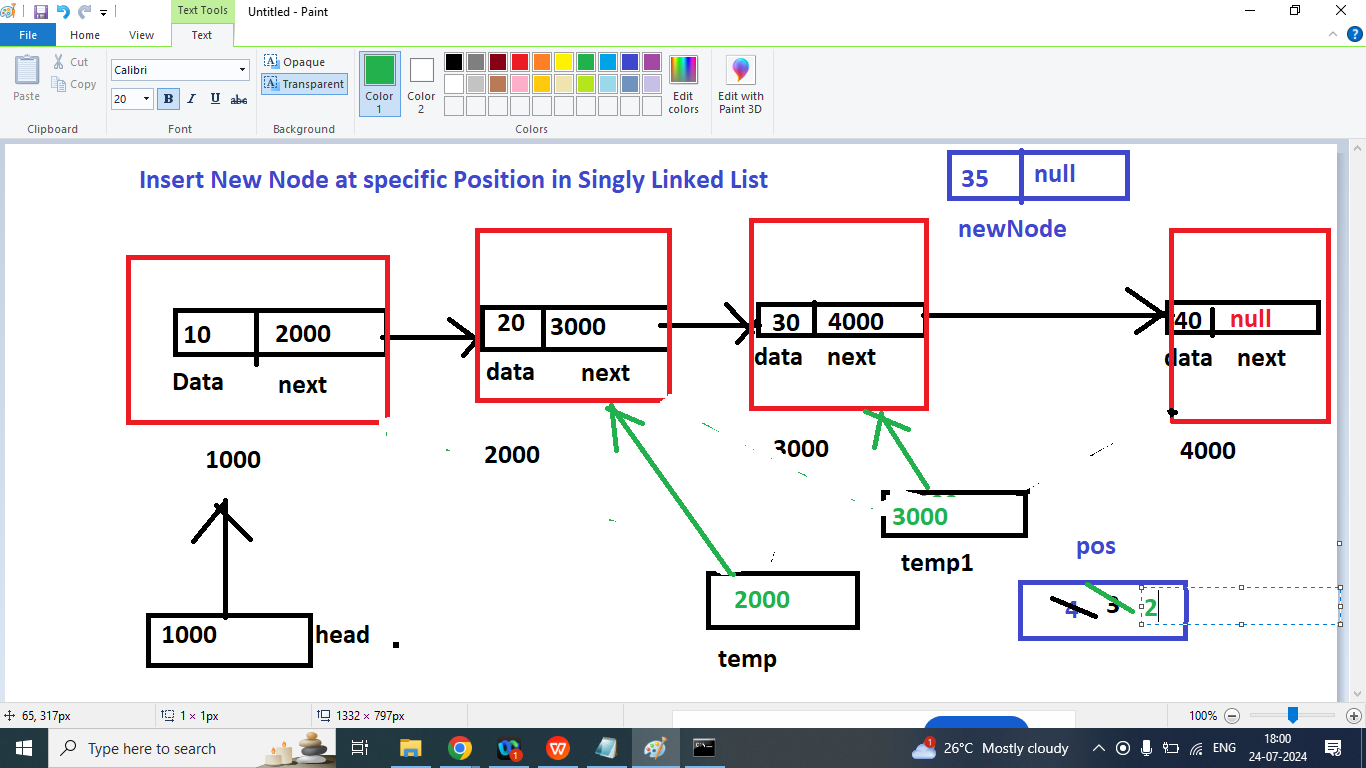




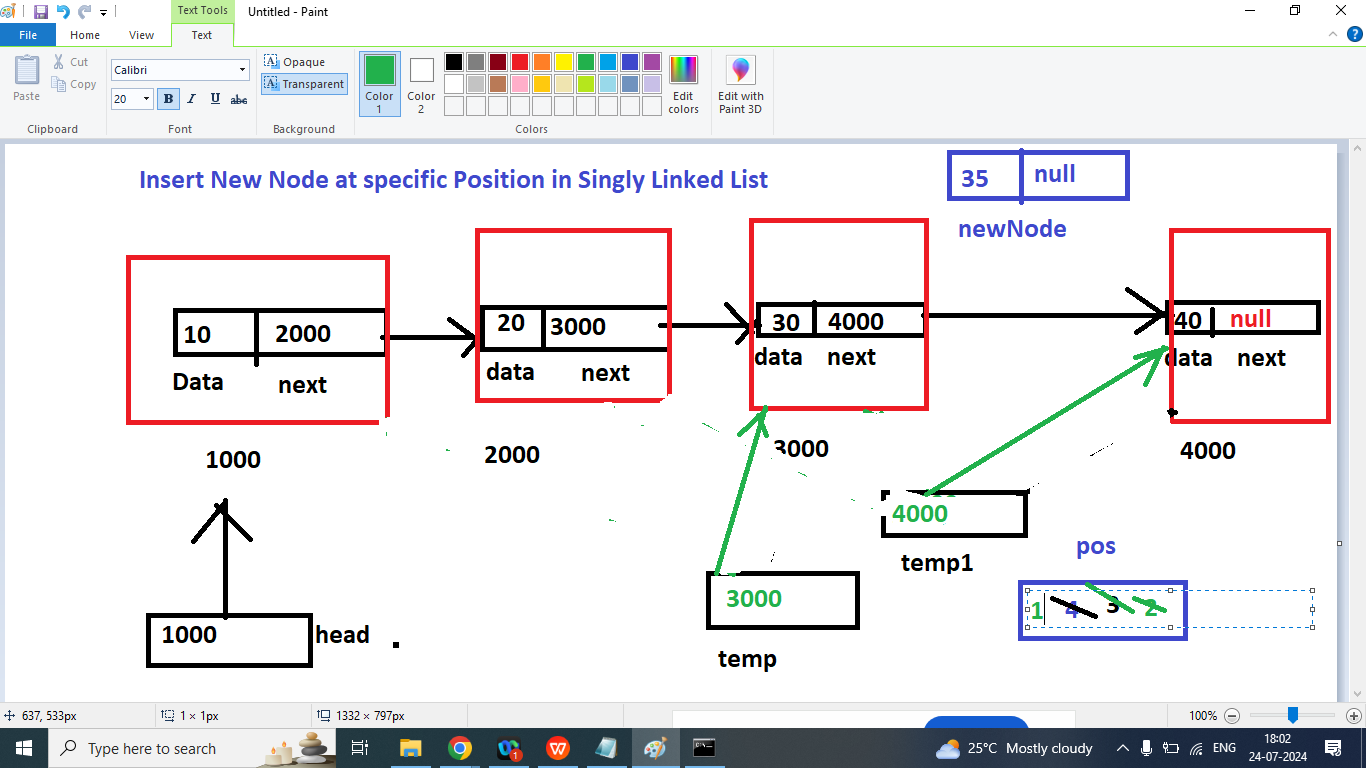
Step1

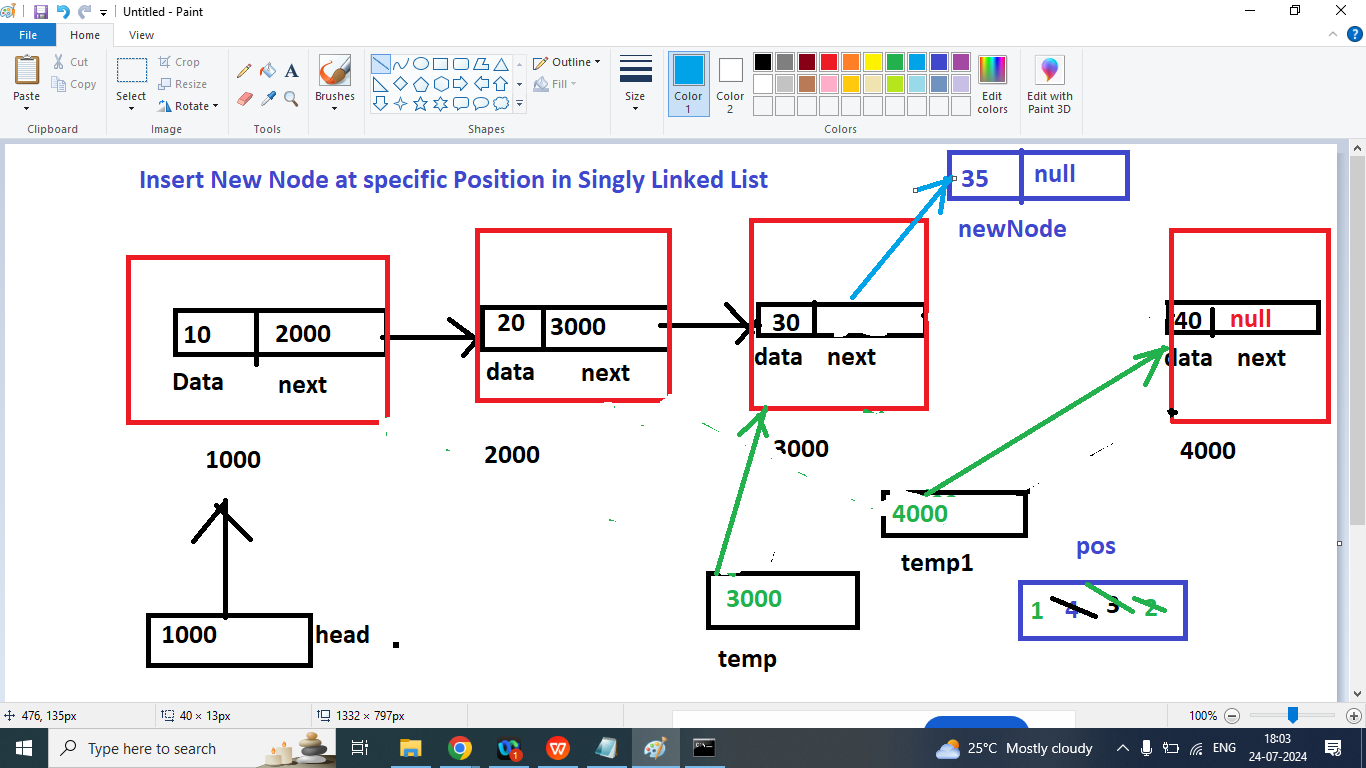


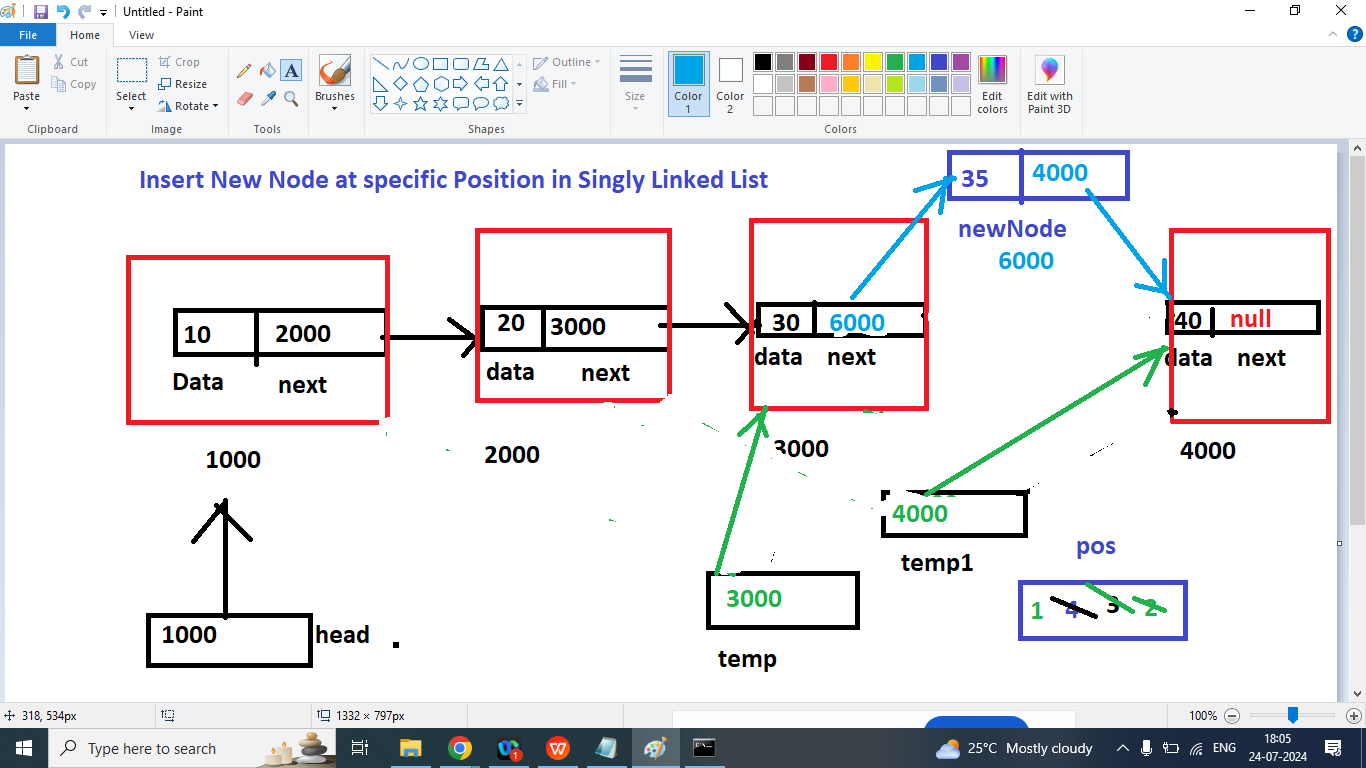
Step2



Step3:







class Node{

int data;

Node next;

public Node(int data){

this.data=data;

this.next=null;

}

public void printData(Node head){

if(head==null){

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

}

else{

Node temp=head;

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

while(temp!=null){

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

temp=temp.next;

}

}

}

public Node addAtStart(Node head,int d){

Node newNode=new Node(d);

newNode.next=head;

head=newNode;

return head;

}

public Node addAtEnd(Node head,int d){

Node newNode=new Node(d);

if(head==null){

head=newNode;

return head;

}else{

Node temp=head;

while(temp.next!=null){

temp=temp.next;

}

temp.next=newNode;

}

return head;

}

public static void main(String args[]){

Node first=new Node(10);

Node second=new Node(20);

Node third=new Node(30);

Node head=first;

head.next=second;

head.next.next=third;

head.printData(head);

System.out.println("\nPrint Data after add new Node at end");

head=head.addAtEnd(head,40);

head.printData(head);

head.insertNodeAtPos(head,35,4);

System.out.println("\nPrint Data after add new Node at Position 4");

head.printData(head);

}

public void insertNodeAtPos(Node head,int data,int pos){

Node newNode=new Node(data);

pos--;

Node temp=head;

Node temp1=head.next;

while(pos>1){

temp=temp.next;

temp1=temp1.next;

pos--;

}

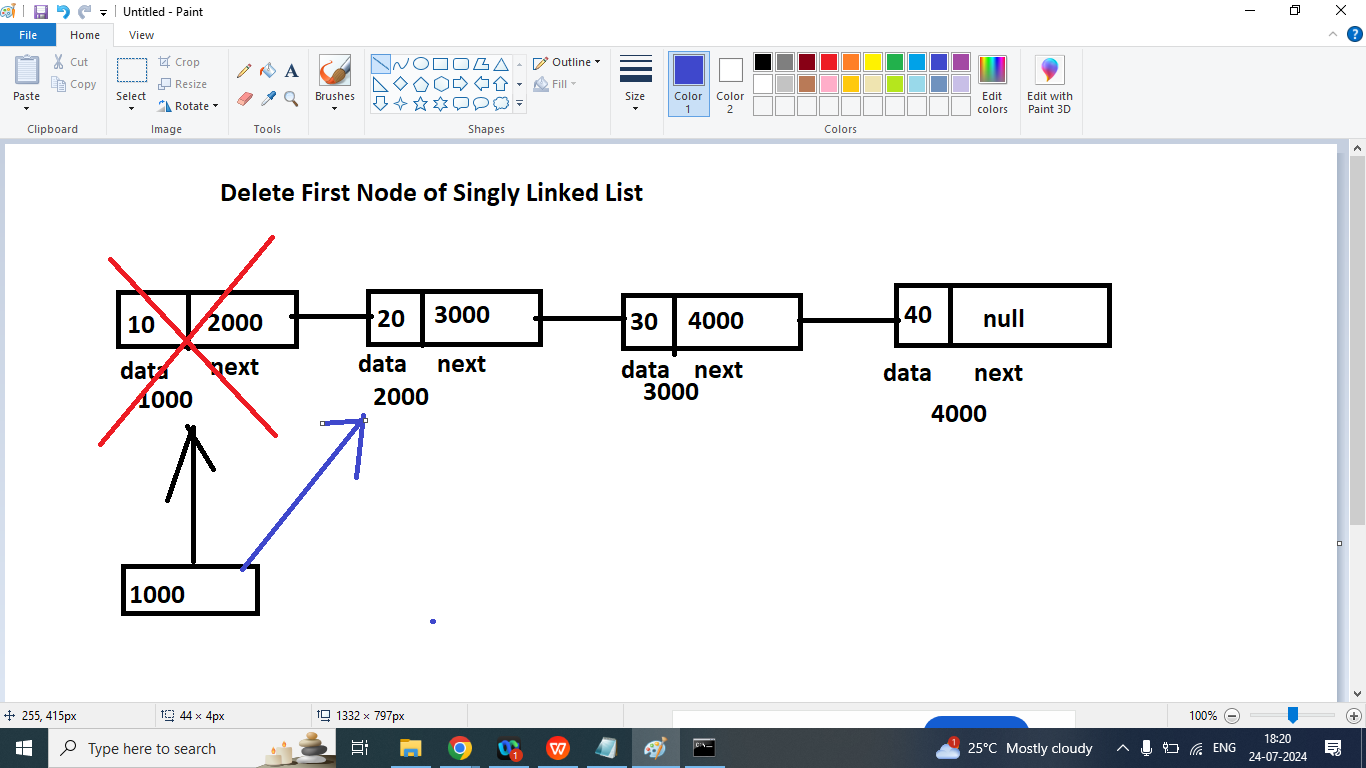
temp.next=newNode;

newNode.next=temp1;

}

}

Q2. Write a java program to delete first Node of Singly Linked List?



class Node{

int data;

Node next;

public Node(int data){

this.data=data;

this.next=null;

}

public void printData(Node head){

if(head==null){

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

}

else{

Node temp=head;

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

while(temp!=null){

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

temp=temp.next;

}

}

}

public Node addAtStart(Node head,int d){

Node newNode=new Node(d);

newNode.next=head;

head=newNode;

return head;

}

public Node addAtEnd(Node head,int d){

Node newNode=new Node(d);

if(head==null){

head=newNode;

return head;

}else{

Node temp=head;

while(temp.next!=null){

temp=temp.next;

}

temp.next=newNode;

}

return head;

}

public static void main(String args[]){

Node first=new Node(10);

Node second=new Node(20);

Node third=new Node(30);

Node head=first;

head.next=second;

head.next.next=third;

head.printData(head);

head=head.addAtEnd(head,40);

head.printData(head);

head=head.deleteFirst(head);

System.out.println("\nPrint Data after delete First Node \n");

head.printData(head);

}

public Node deleteFirst(Node head){

Node temp=head;

if(head==null){

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

}

else{

temp=head.next;

}

return temp;

}

public void insertNodeAtPos(Node head,int data,int pos){

Node newNode=new Node(data);

pos--;

Node temp=head;

Node temp1=head.next;

while(pos>1){

temp=temp.next;

temp1=temp1.next;

pos--;

}

temp.next=newNode;

newNode.next=temp1;

}

}