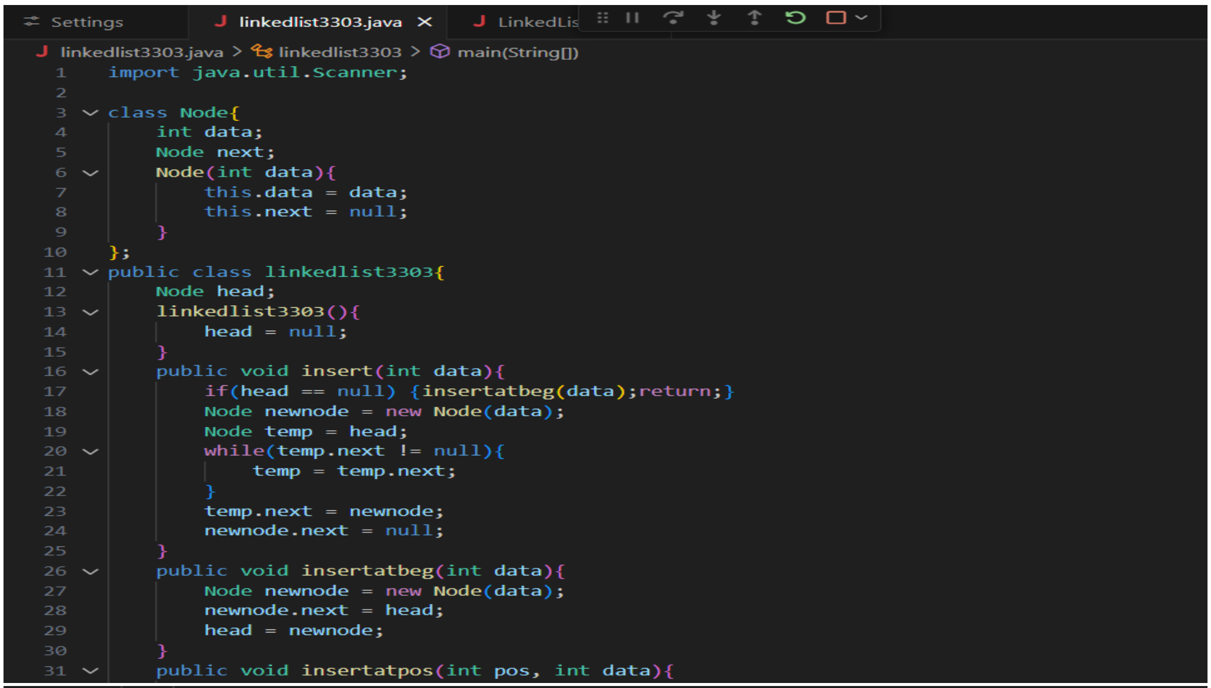
ASSIGNMENT – 3

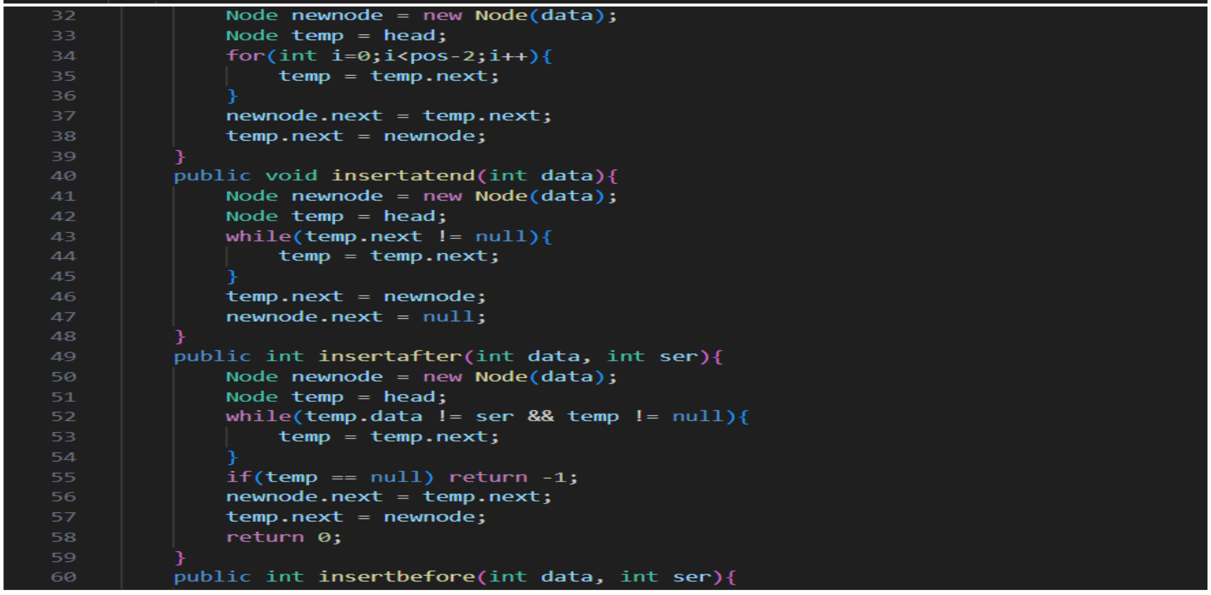
2022503303

QUESTION 1:

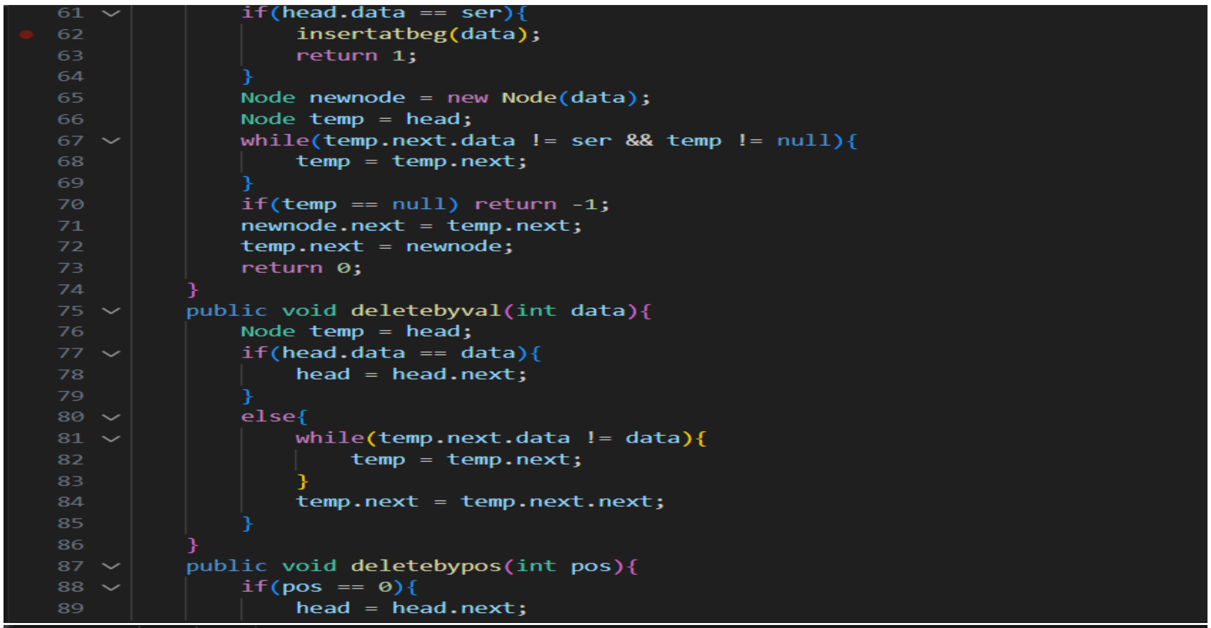
LINKED LIST



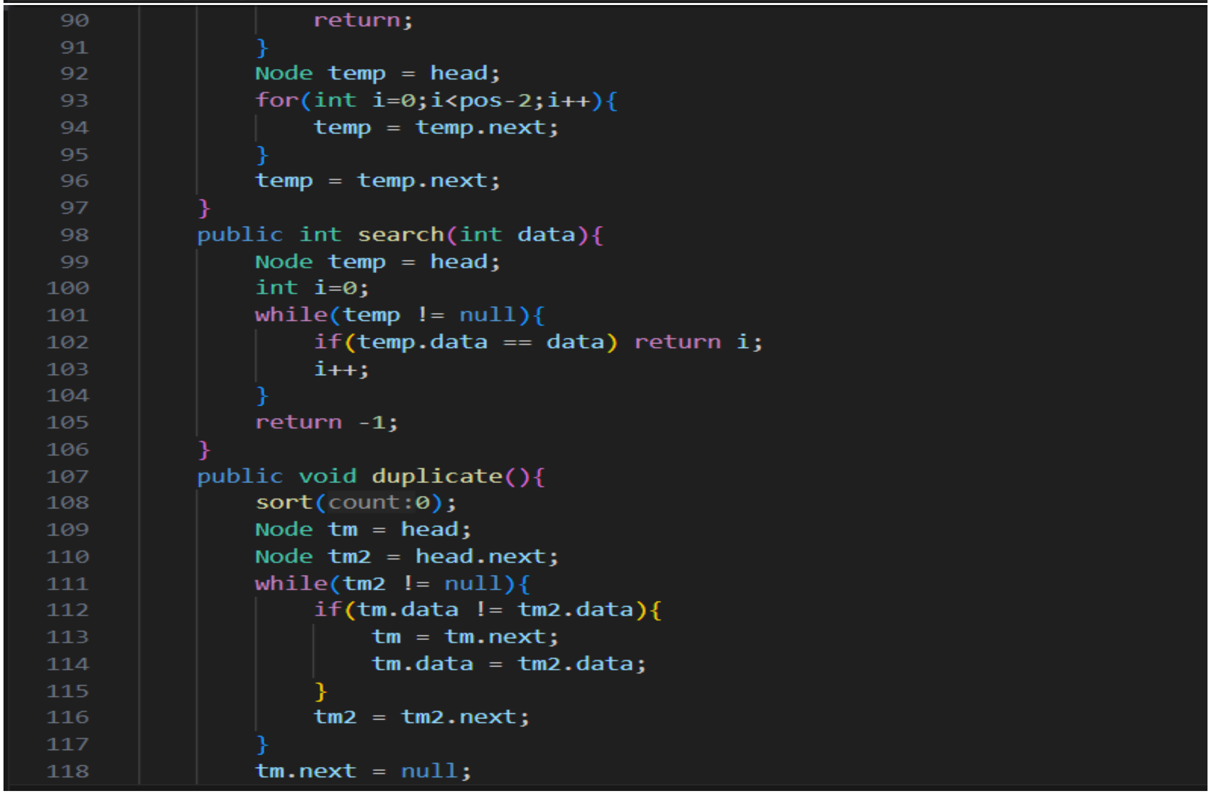
```java
import java.util.Scanner;

class Node{
    int data;
    Node next;
    Node(int data){
        this.data = data;
        this.next = null;
    }
};
public class linkedlist3303{
    Node head;
    linkedlist3303(){
        head = null;
    }
    public void insert(int data){
        if(head == null) {insertatbeg(data);return;}
        Node newnode = new Node(data);
        Node temp = head;
        while(temp.next != null){
            temp = temp.next;
        }
        temp.next = newnode;
        newnode.next = null;
    }
    public void insertatbeg(int data){
        Node newnode = new Node(data);
        newnode.next = head;
        head = newnode;
    }
    public void insertatpos(int pos, int data){
```



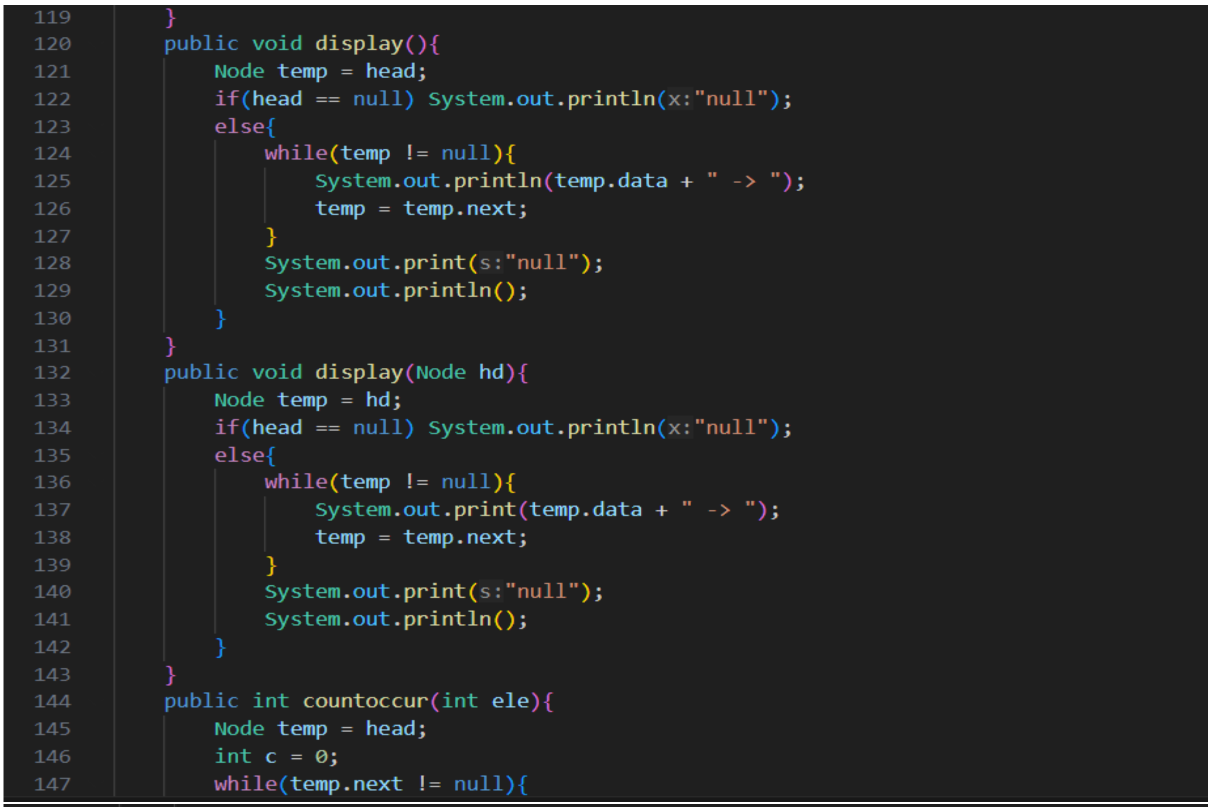
```java
        Node newnode = new Node(data);
        Node temp = head;
        for(int i=0;i<pos-2;i++){
            temp = temp.next;
        }
        newnode.next = temp.next;
        temp.next = newnode;
    }
    public void insertatend(int data){
        Node newnode = new Node(data);
        Node temp = head;
        while(temp.next != null){
            temp = temp.next;
        }
        temp.next = newnode;
        newnode.next = null;
    }
    public int insertafter(int data, int ser){
        Node newnode = new Node(data);
        Node temp = head;
        while(temp.data != ser && temp != null){
            temp = temp.next;
        }
        if(temp == null) return -1;
        newnode.next = temp.next;
        temp.next = newnode;
        return 0;
    }
    public int insertbefore(int data, int ser){
```



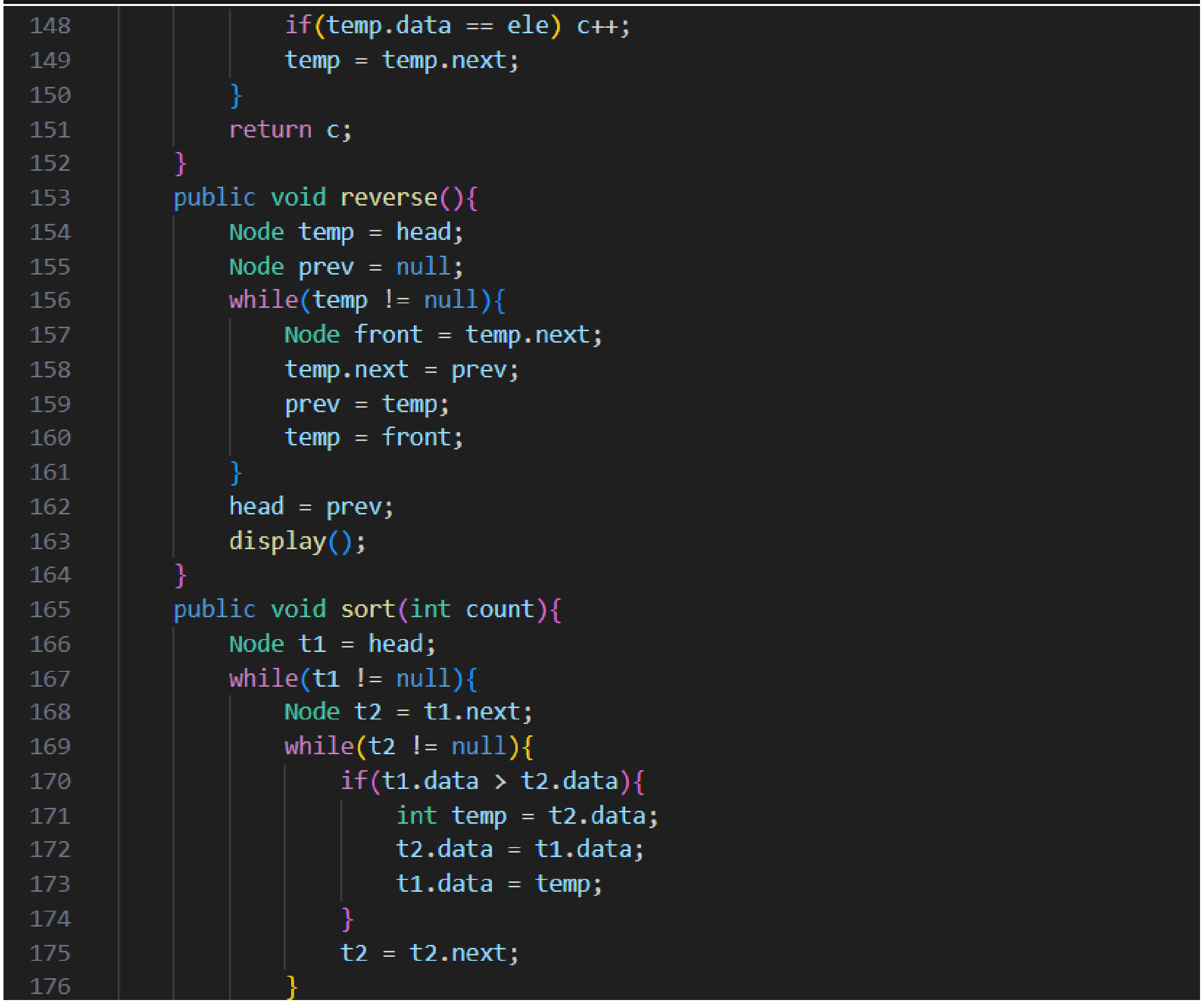
```
        if(head.data == ser){
            insertatbeg(data);
            return 1;
        }
        Node newnode = new Node(data);
        Node temp = head;
        while(temp.next.data != ser && temp != null){
            temp = temp.next;
        }
        if(temp == null) return -1;
        newnode.next = temp.next;
        temp.next = newnode;
        return 0;
    }
    public void deletebyval(int data){
        Node temp = head;
        if(head.data == data){
            head = head.next;
        }
        else{
            while(temp.next.data != data){
                temp = temp.next;
            }
            temp.next = temp.next.next;
        }
    }
    public void deletebypos(int pos){
        if(pos == 0){
            head = head.next;
```



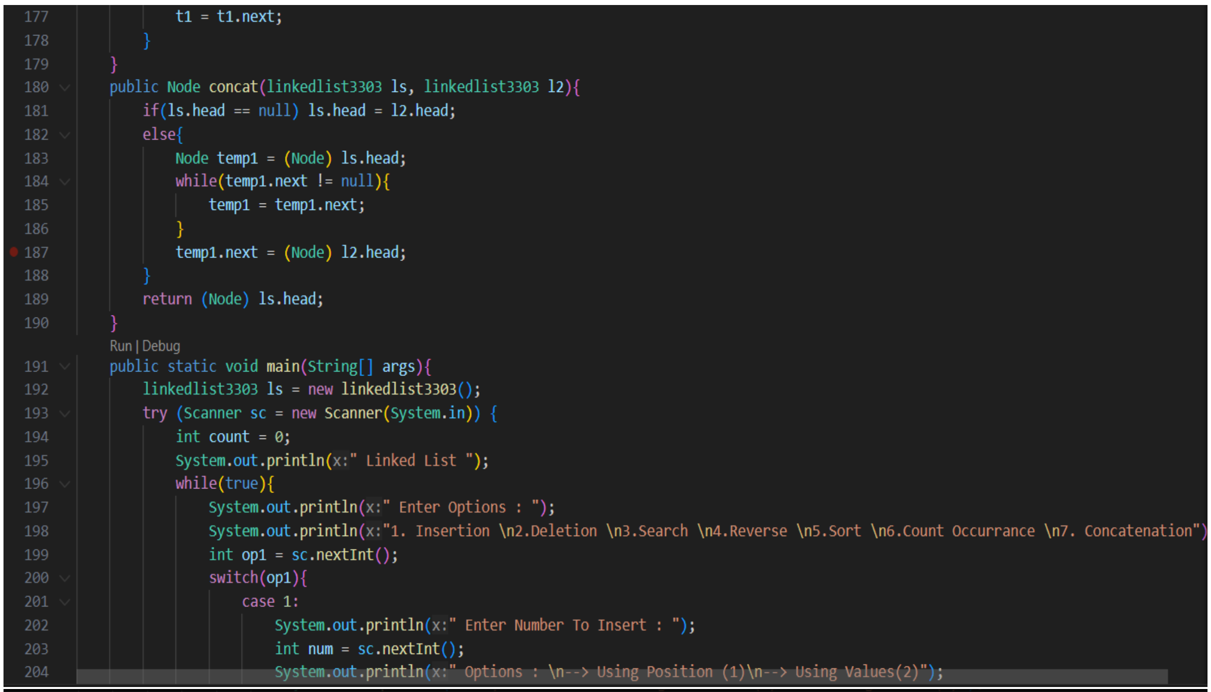
```
            return;
        }
        Node temp = head;
        for(int i=0;i<pos-2;i++){
            temp = temp.next;
        }
        temp = temp.next;
    }
    public int search(int data){
        Node temp = head;
        int i=0;
        while(temp != null){
            if(temp.data == data) return i;
            i++;
        }
        return -1;
    }
    public void duplicate(){
        sort(count:0);
        Node tm = head;
        Node tm2 = head.next;
        while(tm2 != null){
            if(tm.data != tm2.data){
                tm = tm.next;
                tm.data = tm2.data;
            }
            tm2 = tm2.next;
        }
        tm.next = null;
```



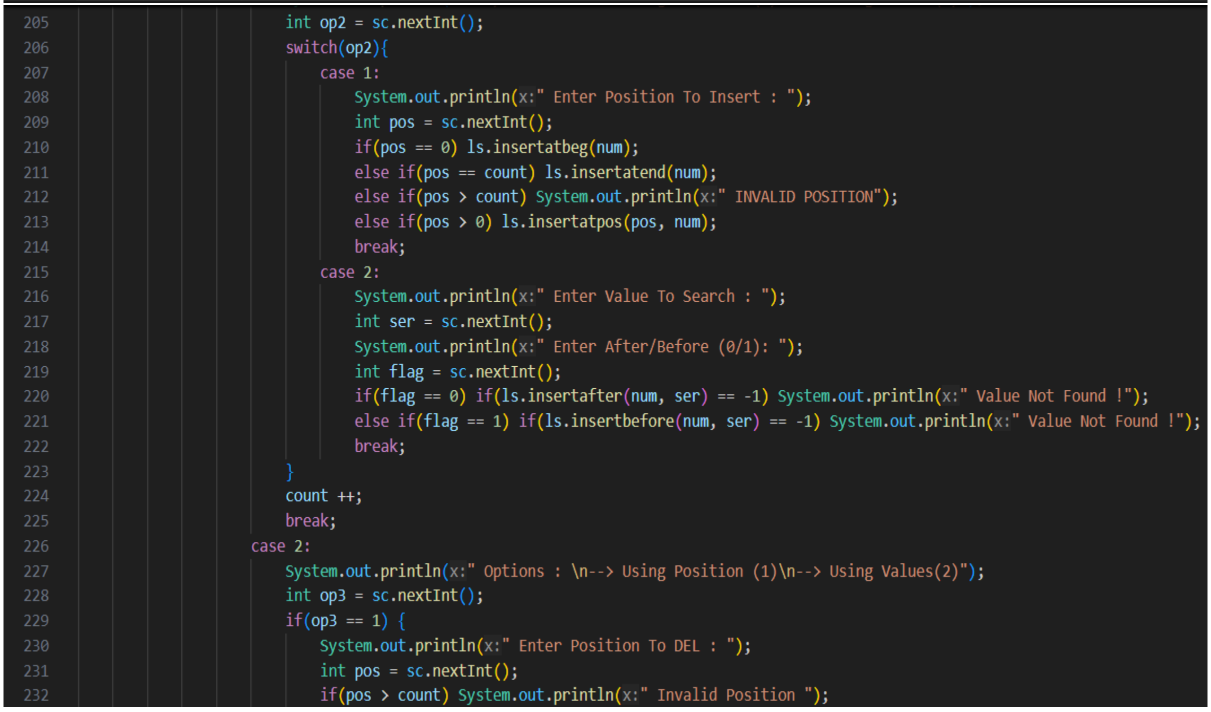
```java
    }
    public void display(){
        Node temp = head;
        if(head == null) System.out.println(x:"null");
        else{
            while(temp != null){
                System.out.println(temp.data + " -> ");
                temp = temp.next;
            }
            System.out.print(s:"null");
            System.out.println();
        }
    }
    public void display(Node hd){
        Node temp = hd;
        if(head == null) System.out.println(x:"null");
        else{
            while(temp != null){
                System.out.print(temp.data + " -> ");
                temp = temp.next;
            }
            System.out.print(s:"null");
            System.out.println();
        }
    }
    public int countoccur(int ele){
        Node temp = head;
        int c = 0;
        while(temp.next != null){
```



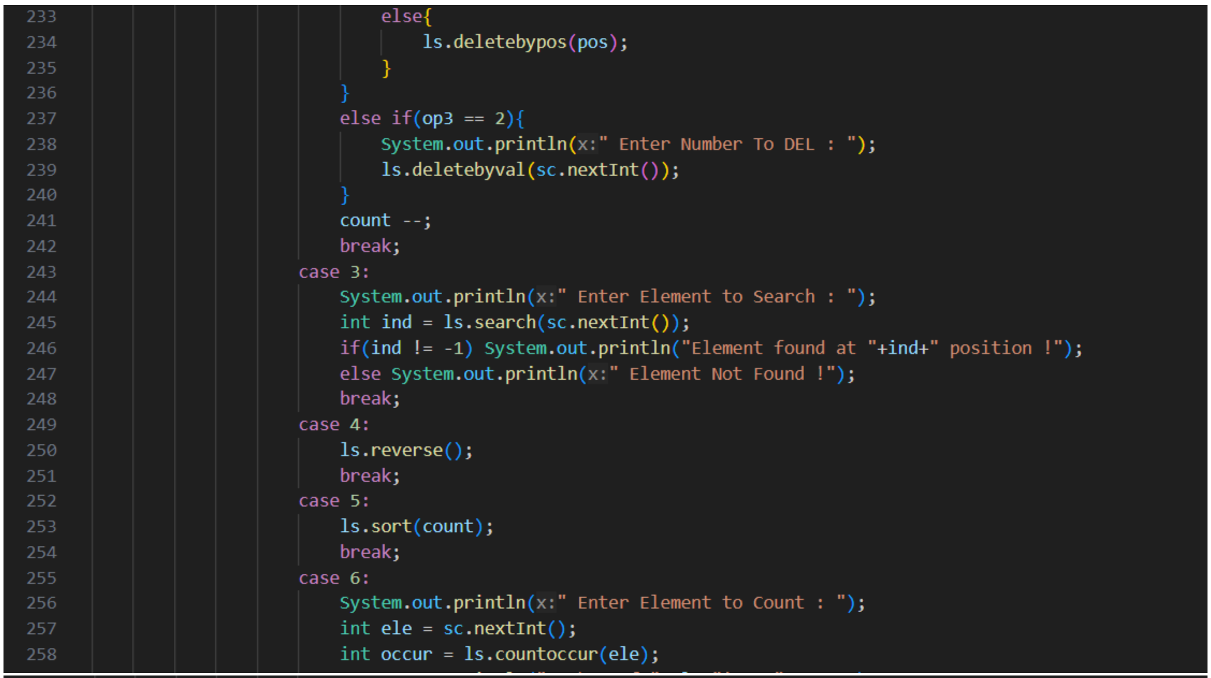
```java
            if(temp.data == ele) c++;
            temp = temp.next;
        }
        return c;
    }
    public void reverse(){
        Node temp = head;
        Node prev = null;
        while(temp != null){
            Node front = temp.next;
            temp.next = prev;
            prev = temp;
            temp = front;
        }
        head = prev;
        display();
    }
    public void sort(int count){
        Node t1 = head;
        while(t1 != null){
            Node t2 = t1.next;
            while(t2 != null){
                if(t1.data > t2.data){
                    int temp = t2.data;
                    t2.data = t1.data;
                    t1.data = temp;
                }
                t2 = t2.next;
            }
```



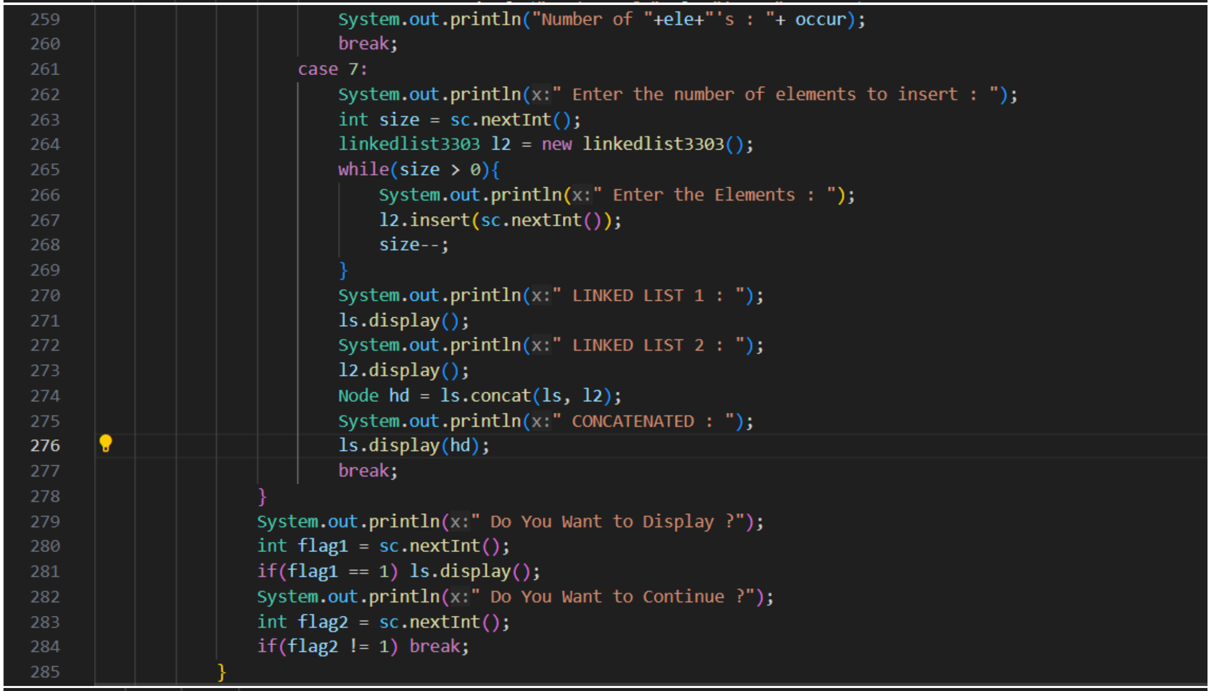
```java
            t1 = t1.next;
        }
    }
    public Node concat(linkedlist3303 ls, linkedlist3303 l2){
        if(ls.head == null) ls.head = l2.head;
        else{
            Node temp1 = (Node) ls.head;
            while(temp1.next != null){
                temp1 = temp1.next;
            }
            temp1.next = (Node) l2.head;
        }
        return (Node) ls.head;
    }
    Run | Debug
    public static void main(String[] args){
        linkedlist3303 ls = new linkedlist3303();
        try (Scanner sc = new Scanner(System.in)) {
            int count = 0;
            System.out.println(x:" Linked List ");
            while(true){
                System.out.println(x:" Enter Options : ");
                System.out.println(x:"1. Insertion \n2.Deletion \n3.Search \n4.Reverse \n5.Sort \n6.Count Occurrance \n7. Concatenation")
                int op1 = sc.nextInt();
                switch(op1){
                    case 1:
                        System.out.println(x:" Enter Number To Insert : ");
                        int num = sc.nextInt();
                        System.out.println(x:" Options : \n--> Using Position (1)\n--> Using Values(2)");
```



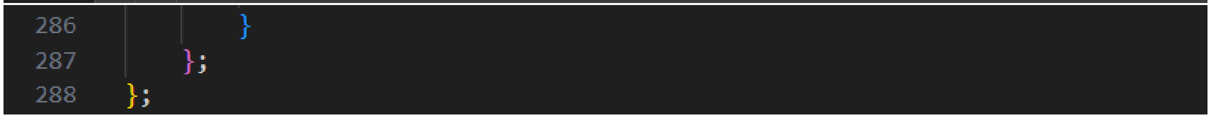
```java
                        int op2 = sc.nextInt();
                        switch(op2){
                            case 1:
                                System.out.println(x:" Enter Position To Insert : ");
                                int pos = sc.nextInt();
                                if(pos == 0) ls.insertatbeg(num);
                                else if(pos == count) ls.insertatend(num);
                                else if(pos > count) System.out.println(x:" INVALID POSITION");
                                else if(pos > 0) ls.insertatpos(pos, num);
                                break;
                            case 2:
                                System.out.println(x:" Enter Value To Search : ");
                                int ser = sc.nextInt();
                                System.out.println(x:" Enter After/Before (0/1): ");
                                int flag = sc.nextInt();
                                if(flag == 0) if(ls.insertafter(num, ser) == -1) System.out.println(x:" Value Not Found !");
                                else if(flag == 1) if(ls.insertbefore(num, ser) == -1) System.out.println(x:" Value Not Found !");
                                break;
                        }
                        count ++;
                        break;
                    case 2:
                        System.out.println(x:" Options : \n--> Using Position (1)\n--> Using Values(2)");
                        int op3 = sc.nextInt();
                        if(op3 == 1) {
                            System.out.println(x:" Enter Position To DEL : ");
                            int pos = sc.nextInt();
                            if(pos > count) System.out.println(x:" Invalid Position ");
```



```java
                            else{
                                ls.deletebypos(pos);
                            }
                        }
                        else if(op3 == 2){
                            System.out.println(x:" Enter Number To DEL : ");
                            ls.deletebyval(sc.nextInt());
                        }
                        count --;
                        break;
                    case 3:
                        System.out.println(x:" Enter Element to Search : ");
                        int ind = ls.search(sc.nextInt());
                        if(ind != -1) System.out.println("Element found at "+ind+" position !");
                        else System.out.println(x:" Element Not Found !");
                        break;
                    case 4:
                        ls.reverse();
                        break;
                    case 5:
                        ls.sort(count);
                        break;
                    case 6:
                        System.out.println(x:" Enter Element to Count : ");
                        int ele = sc.nextInt();
                        int occur = ls.countoccur(ele);
```



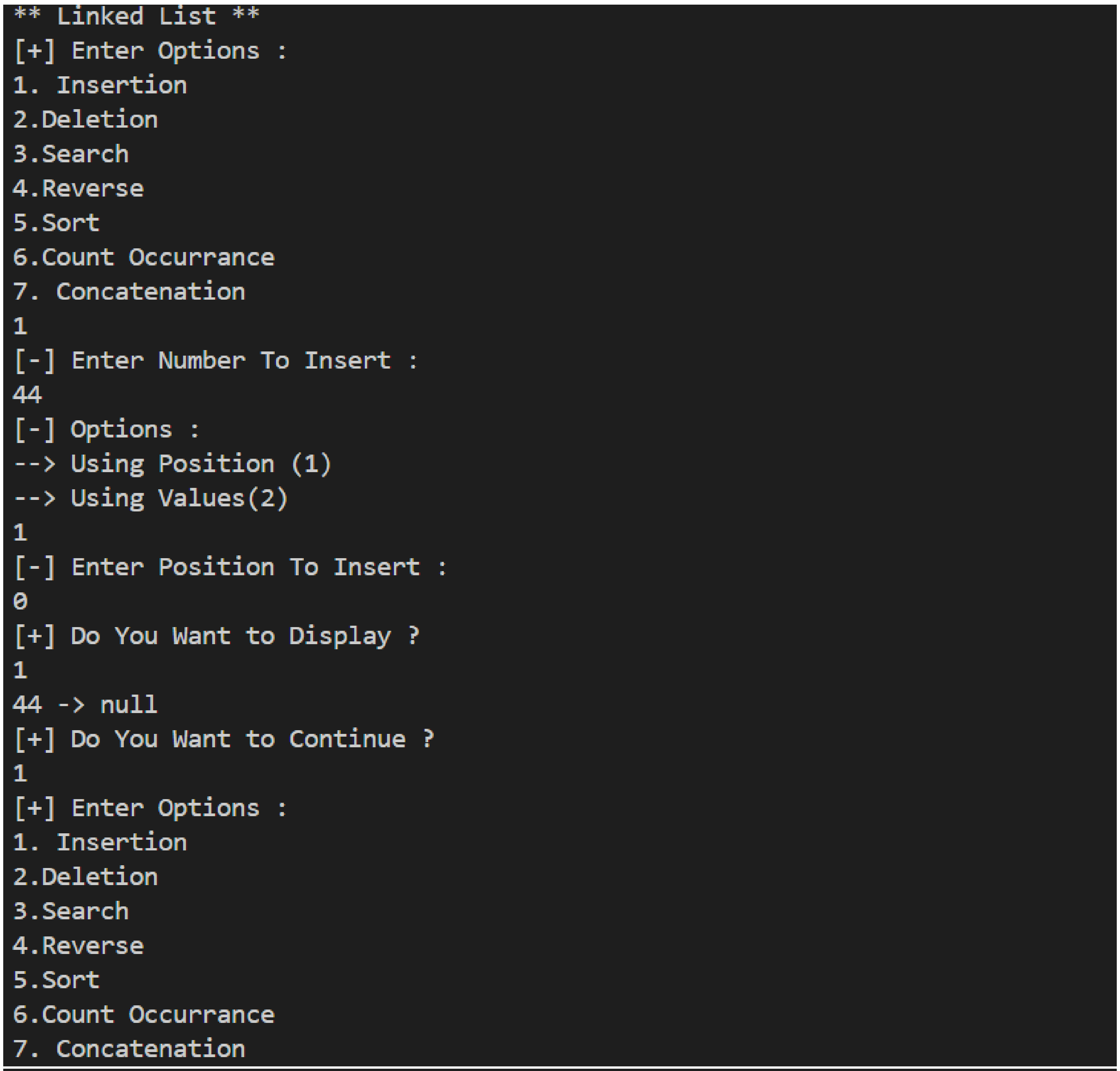
```java
                        System.out.println("Number of "+ele+"'s : "+ occur);
                        break;
                    case 7:
                        System.out.println(x:" Enter the number of elements to insert : ");
                        int size = sc.nextInt();
                        linkedlist3303 l2 = new linkedlist3303();
                        while(size > 0){
                            System.out.println(x:" Enter the Elements : ");
                            l2.insert(sc.nextInt());
                            size--;
                        }
                        System.out.println(x:" LINKED LIST 1 : ");
                        ls.display();
                        System.out.println(x:" LINKED LIST 2 : ");
                        l2.display();
                        Node hd = ls.concat(ls, l2);
                        System.out.println(x:" CONCATENATED : ");
                        ls.display(hd);
                        break;
                }
                System.out.println(x:" Do You Want to Display ?");
                int flag1 = sc.nextInt();
                if(flag1 == 1) ls.display();
                System.out.println(x:" Do You Want to Continue ?");
                int flag2 = sc.nextInt();
                if(flag2 != 1) break;
            }
```



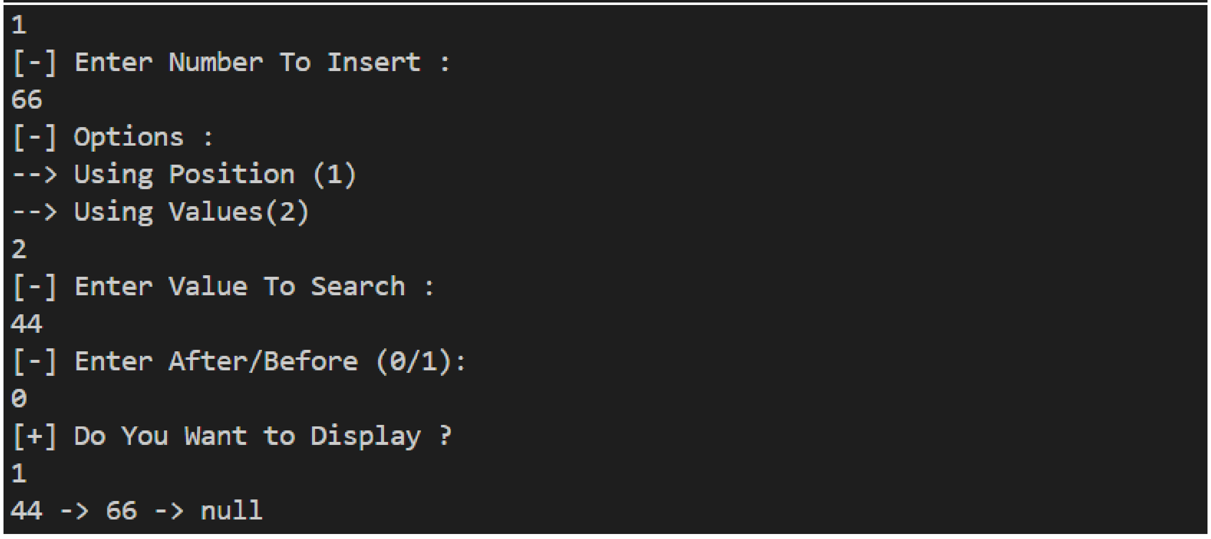
```java
        }
    };
};
```

OUTPUT:

Insertion

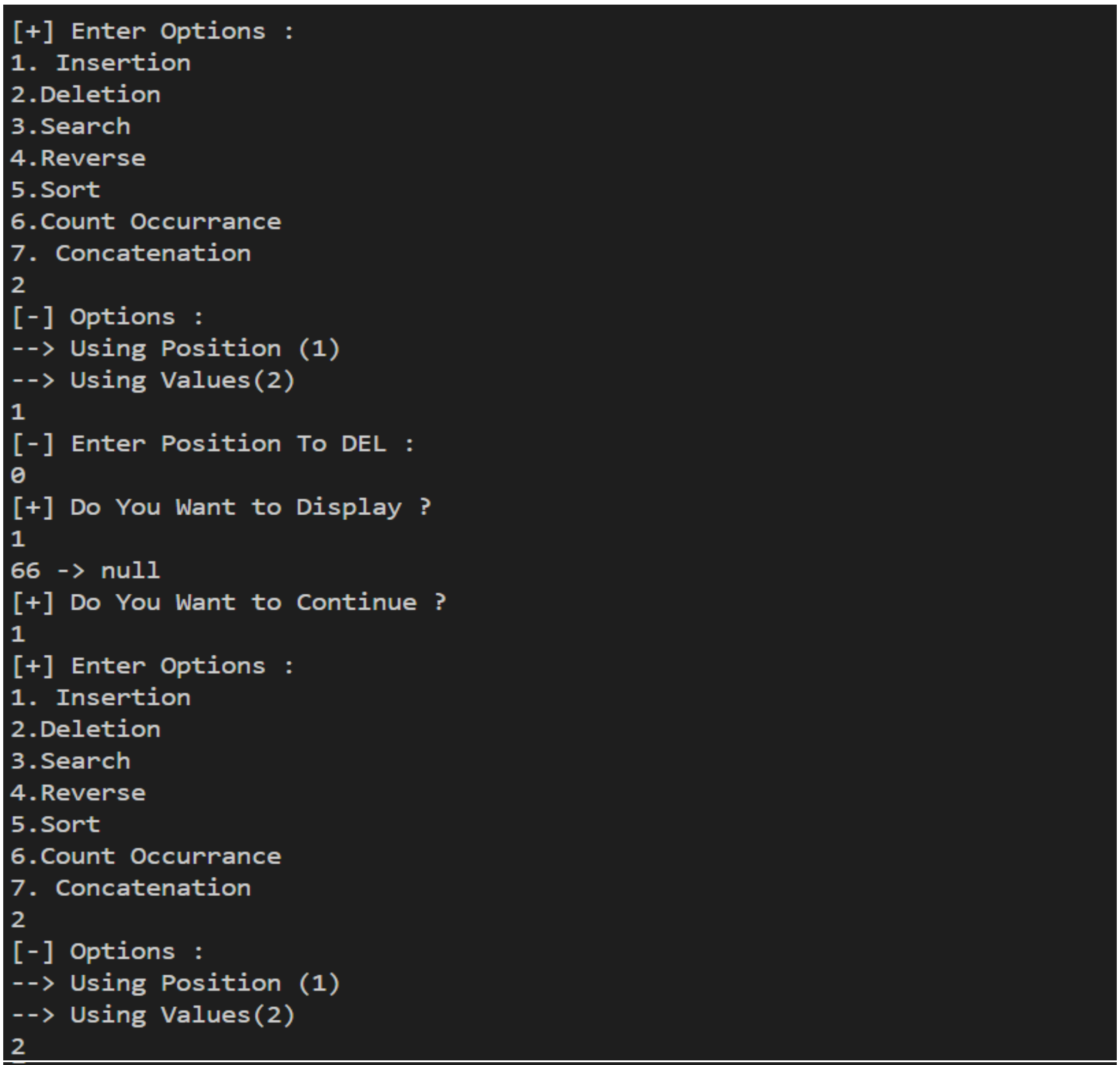


```
** Linked List **
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
1
[-] Enter Number To Insert :
44
[-] Options :
--> Using Position (1)
--> Using Values(2)
1
[-] Enter Position To Insert :
0
[+] Do You Want to Display ?
1
44 -> null
[+] Do You Want to Continue ?
1
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
```

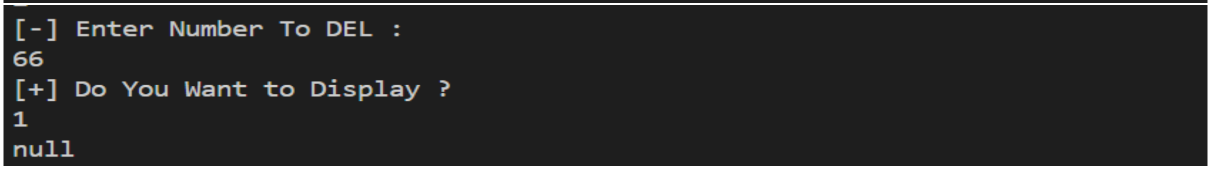


```
1
[-] Enter Number To Insert :
66
[-] Options :
--> Using Position (1)
--> Using Values(2)
2
[-] Enter Value To Search :
44
[-] Enter After/Before (0/1):
0
[+] Do You Want to Display ?
1
44 -> 66 -> null
```

Deletion

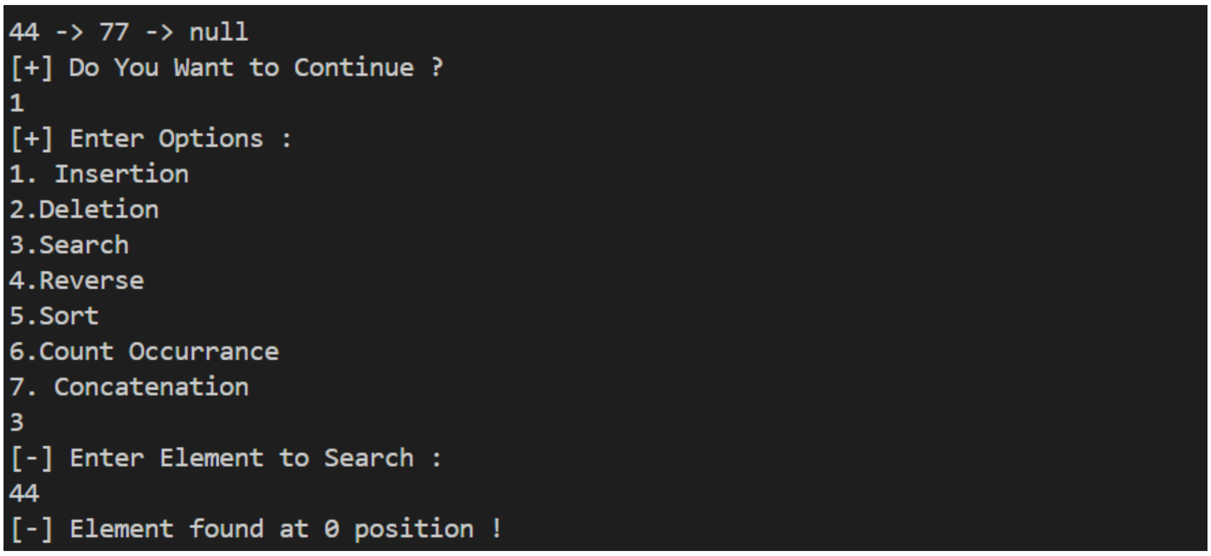


```
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
2
[-] Options :
--> Using Position (1)
--> Using Values(2)
1
[-] Enter Position To DEL :
0
[+] Do You Want to Display ?
1
66 -> null
[+] Do You Want to Continue ?
1
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
2
[-] Options :
--> Using Position (1)
--> Using Values(2)
2
```



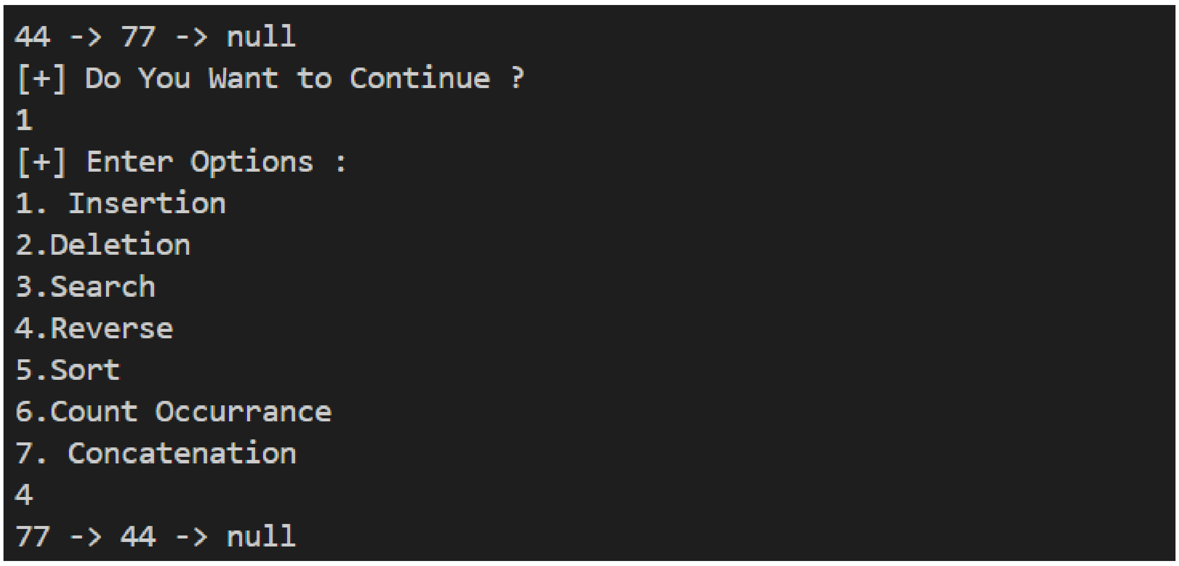
```
[-] Enter Number To DEL :
66
[+] Do You Want to Display ?
1
null
```

Search



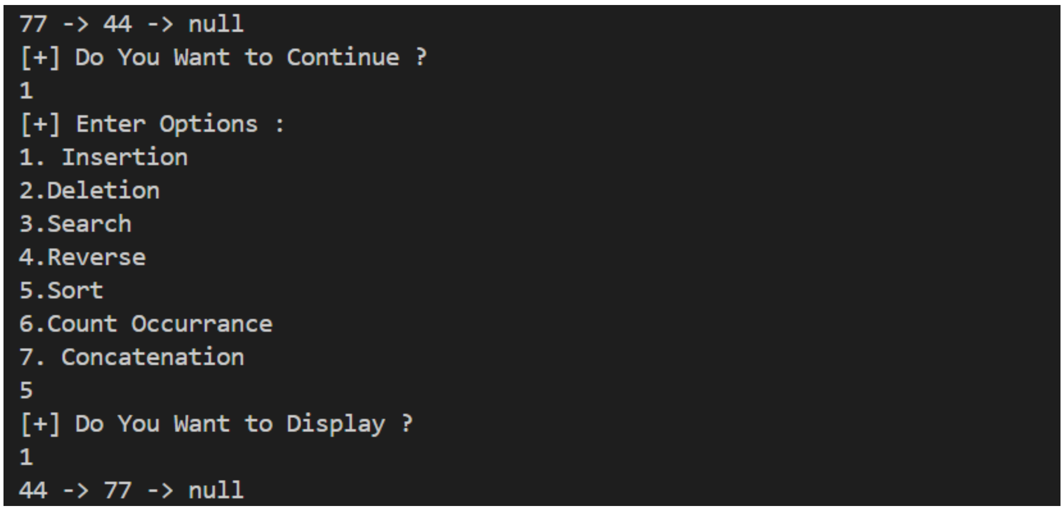
```
44 -> 77 -> null
[+] Do You Want to Continue ?
1
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
3
[-] Enter Element to Search :
44
[-] Element found at 0 position !
```

Reverse



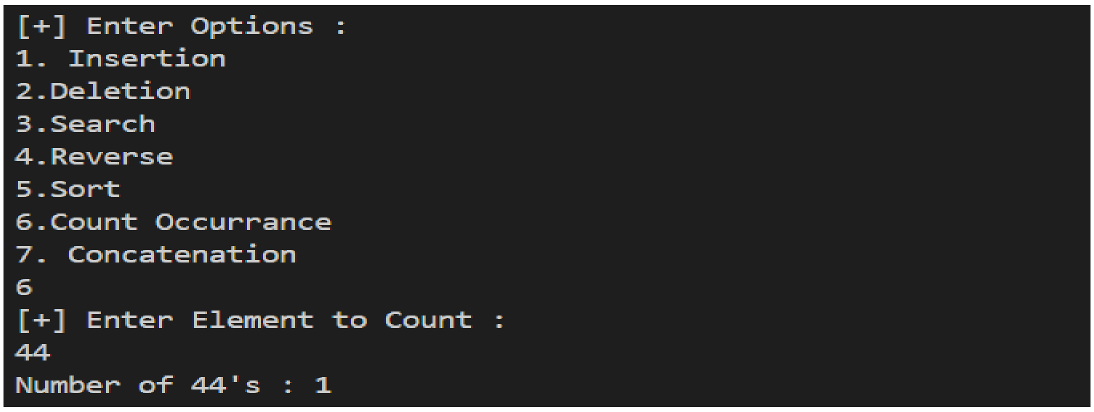
```
44 -> 77 -> null
[+] Do You Want to Continue ?
1
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
4
77 -> 44 -> null
```

Sort



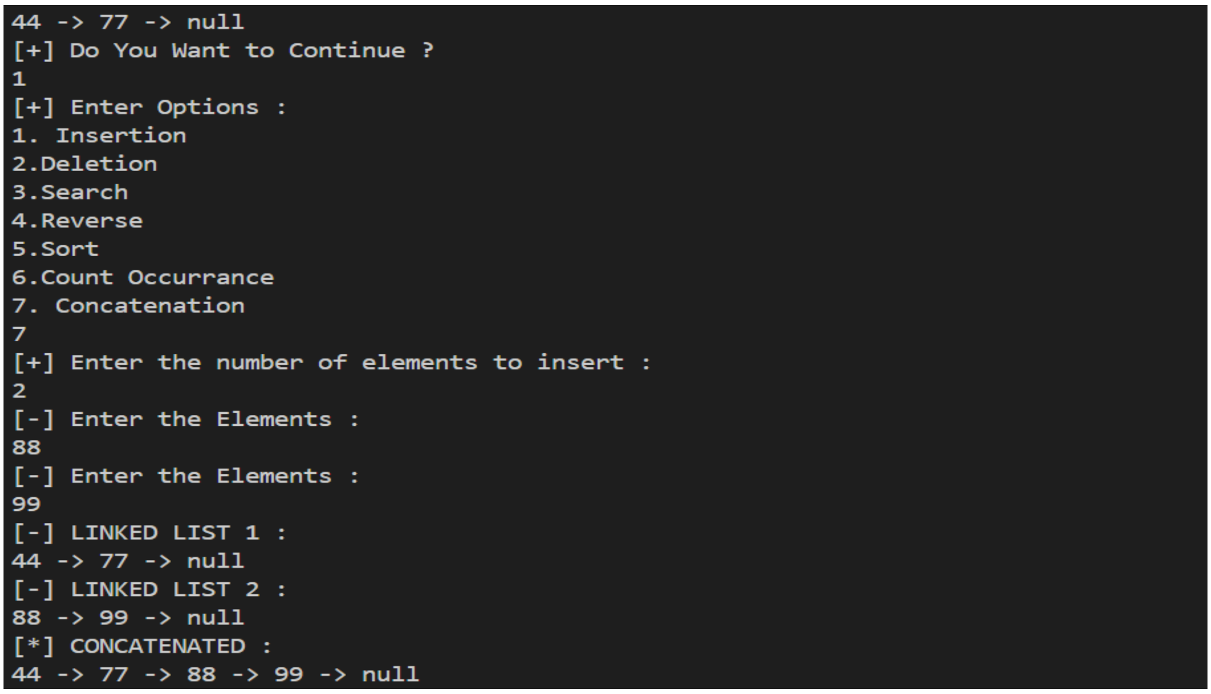
```
77 -> 44 -> null
[+] Do You Want to Continue ?
1
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
5
[+] Do You Want to Display ?
1
44 -> 77 -> null
```

Count Occurance



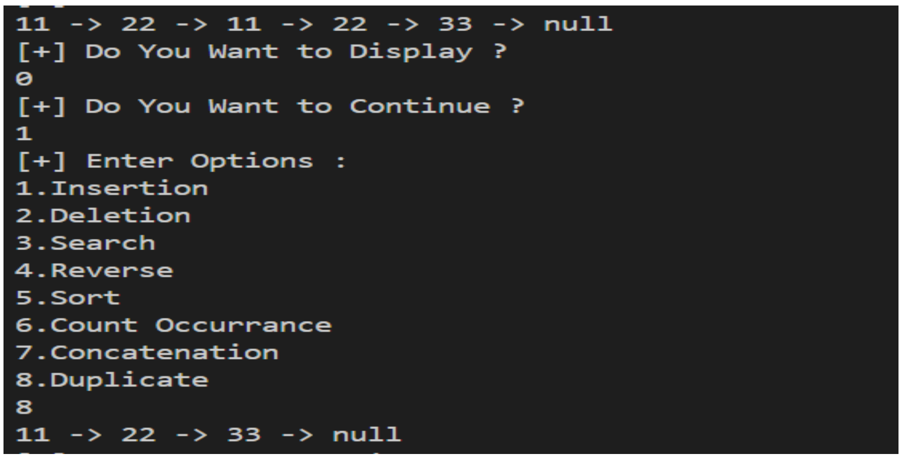
```
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
6
[+] Enter Element to Count :
44
Number of 44's : 1
```

Concatenated



```
44 -> 77 -> null
[+] Do You Want to Continue ?
1
[+] Enter Options :
1. Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7. Concatenation
7
[+] Enter the number of elements to insert :
2
[-] Enter the Elements :
88
[-] Enter the Elements :
99
[-] LINKED LIST 1 :
44 -> 77 -> null
[-] LINKED LIST 2 :
88 -> 99 -> null
[*] CONCATENATED :
44 -> 77 -> 88 -> 99 -> null
```

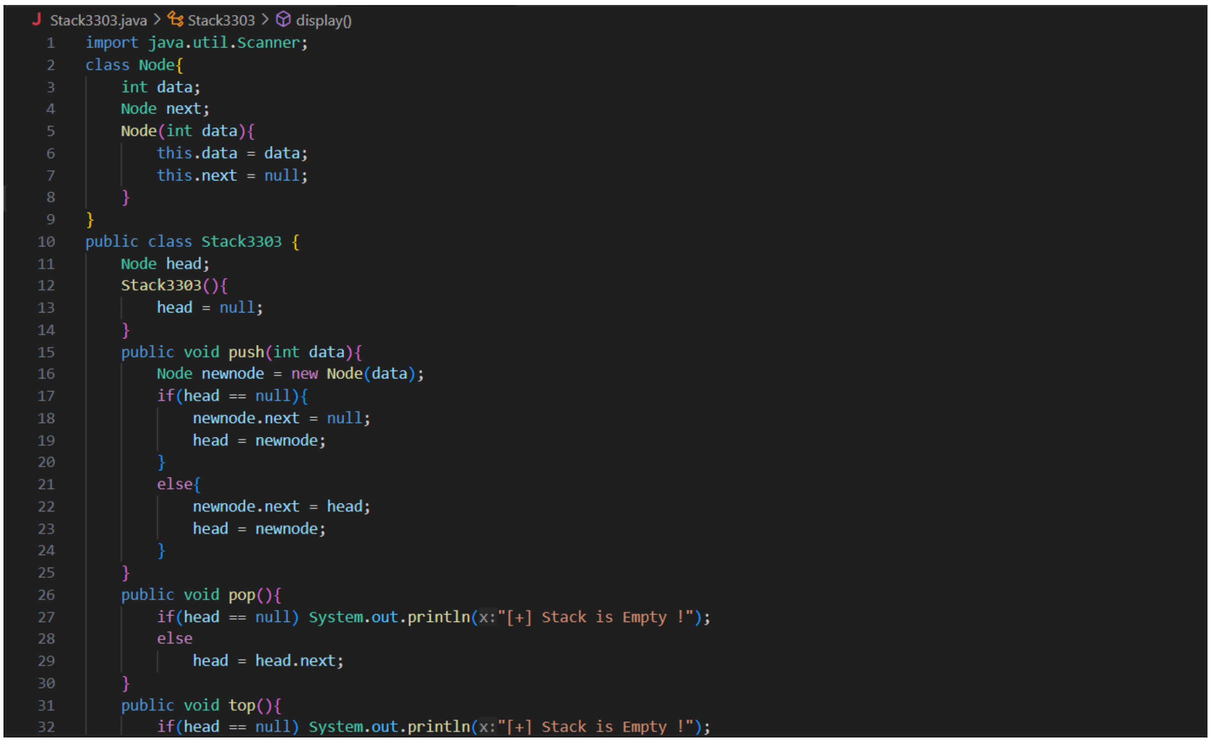
Duplicate



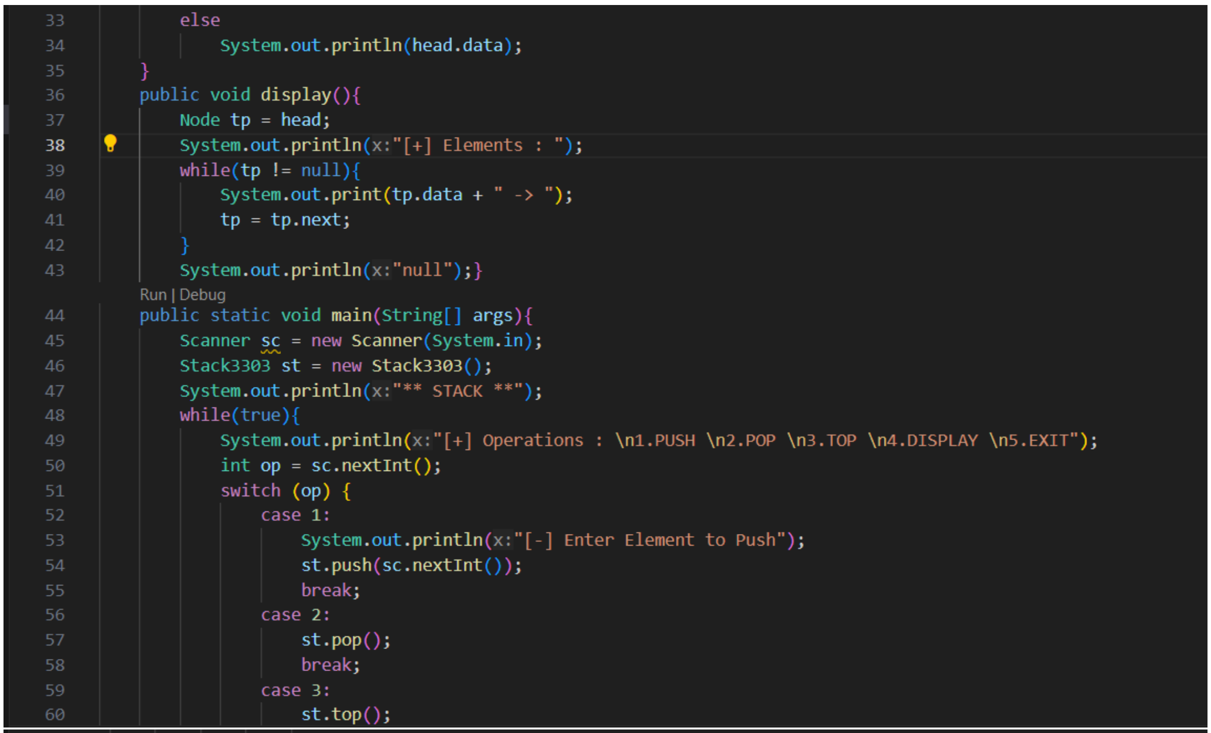
```
11 -> 22 -> 11 -> 22 -> 33 -> null
[+] Do You Want to Display ?
0
[+] Do You Want to Continue ?
1
[+] Enter Options :
1.Insertion
2.Deletion
3.Search
4.Reverse
5.Sort
6.Count Occurrance
7.Concatenation
8.Duplicate
8
11 -> 22 -> 33 -> null
```

QUESTION 2:

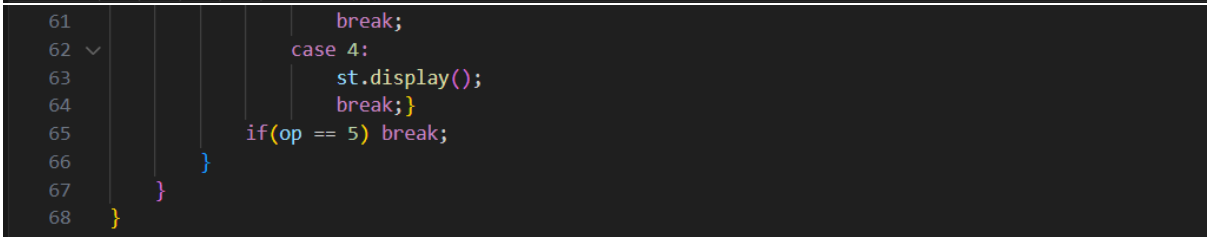
STACK



```java
import java.util.Scanner;
class Node{
    int data;
    Node next;
    Node(int data){
        this.data = data;
        this.next = null;
    }
}
public class Stack3303 {
    Node head;
    Stack3303(){
        head = null;
    }
    public void push(int data){
        Node newnode = new Node(data);
        if(head == null){
            newnode.next = null;
            head = newnode;
        }
        else{
            newnode.next = head;
            head = newnode;
        }
    }
    public void pop(){
        if(head == null) System.out.println(x:"[+] Stack is Empty !");
        else
            head = head.next;
    }
    public void top(){
        if(head == null) System.out.println(x:"[+] Stack is Empty !");
```

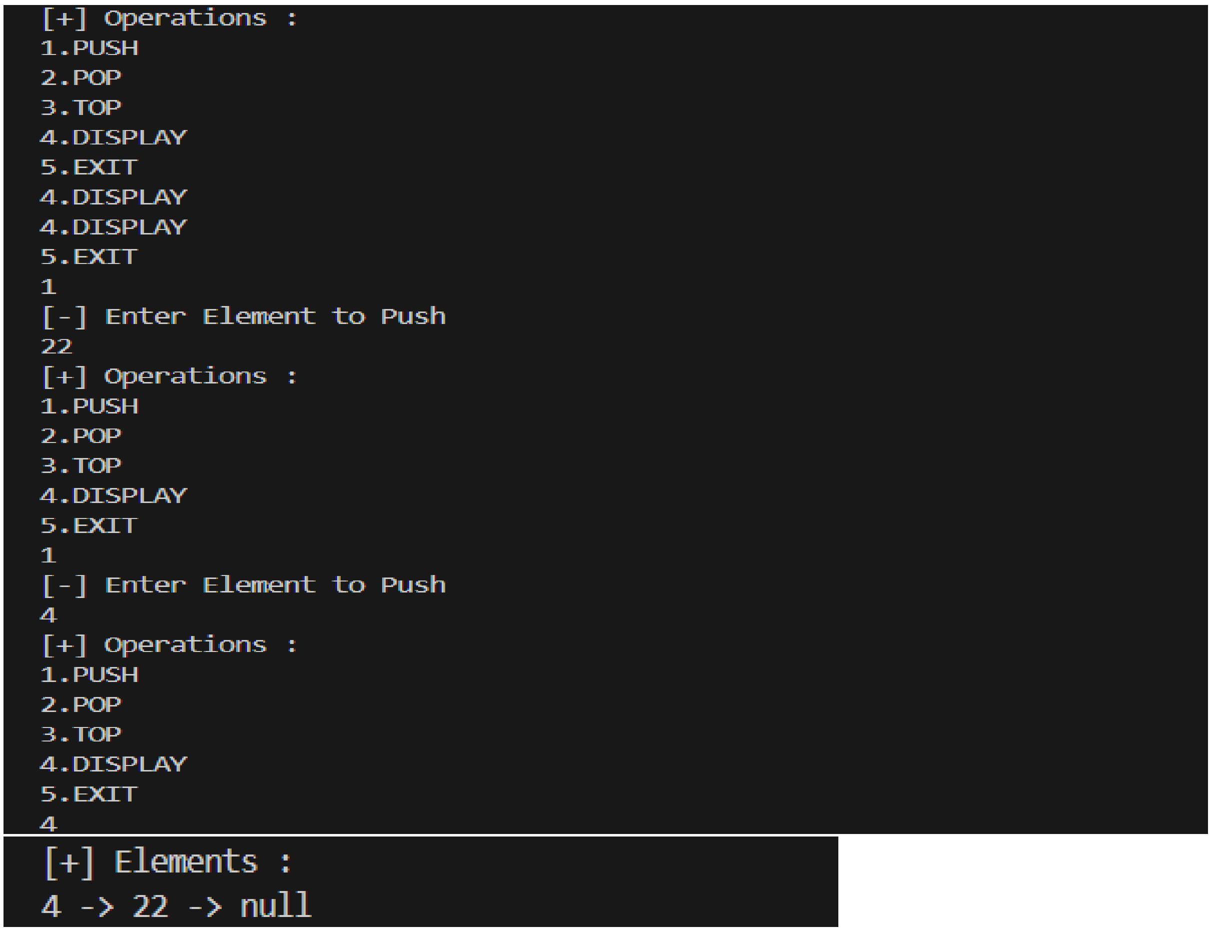


```java
        else
            System.out.println(head.data);
    }
    public void display(){
        Node tp = head;
        System.out.println(x:"[+] Elements : ");
        while(tp != null){
            System.out.print(tp.data + " -> ");
            tp = tp.next;
        }
        System.out.println(x:"null");}
    Run | Debug
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        Stack3303 st = new Stack3303();
        System.out.println(x:"** STACK **");
        while(true){
            System.out.println(x:"[+] Operations : \n1.PUSH \n2.POP \n3.TOP \n4.DISPLAY \n5.EXIT");
            int op = sc.nextInt();
            switch (op) {
                case 1:
                    System.out.println(x:"[-] Enter Element to Push");
                    st.push(sc.nextInt());
                    break;
                case 2:
                    st.pop();
                    break;
                case 3:
                    st.top();
```

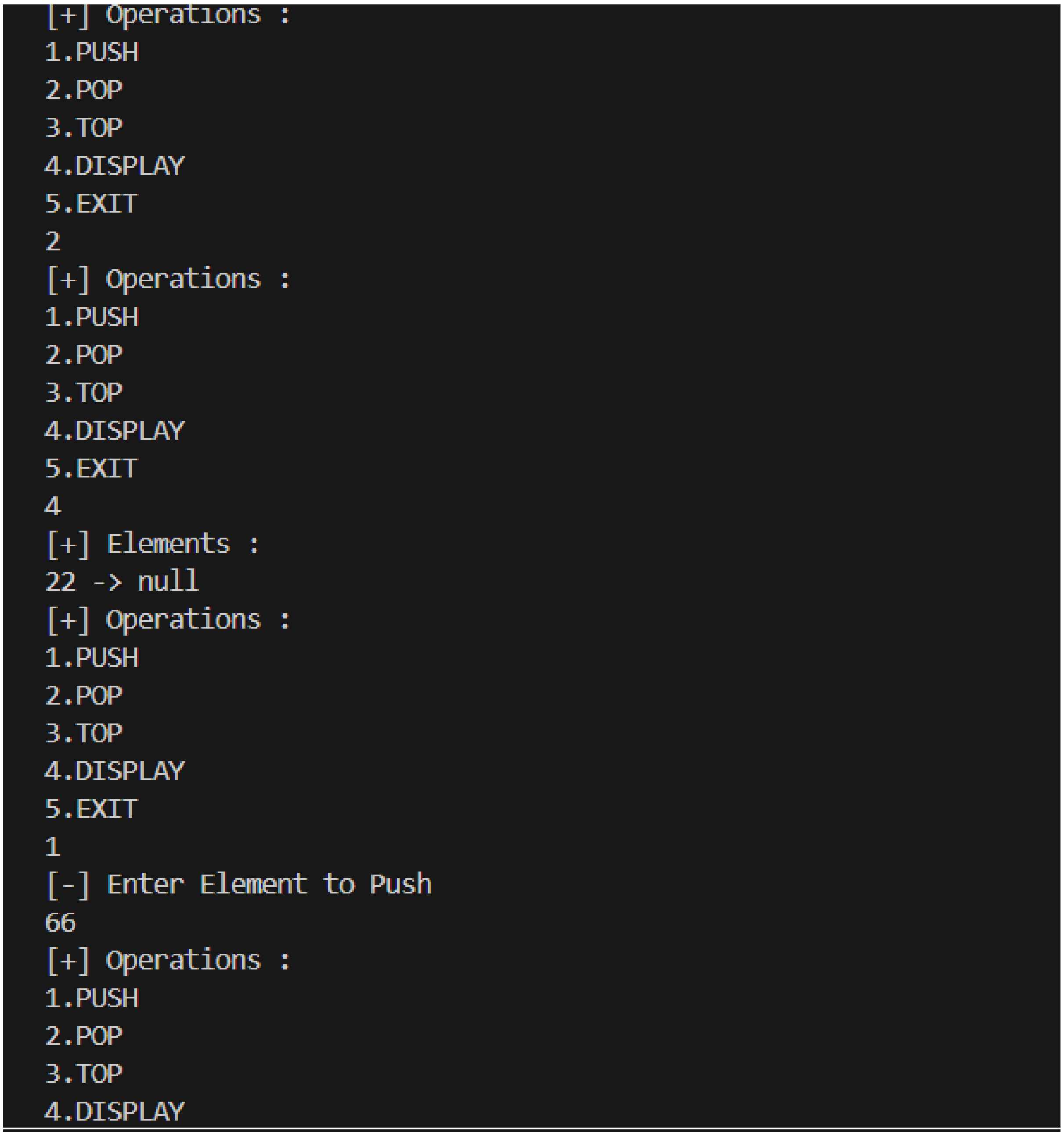


```java
                    break;
                case 4:
                    st.display();
                    break;}
            if(op == 5) break;
        }
    }
}
```

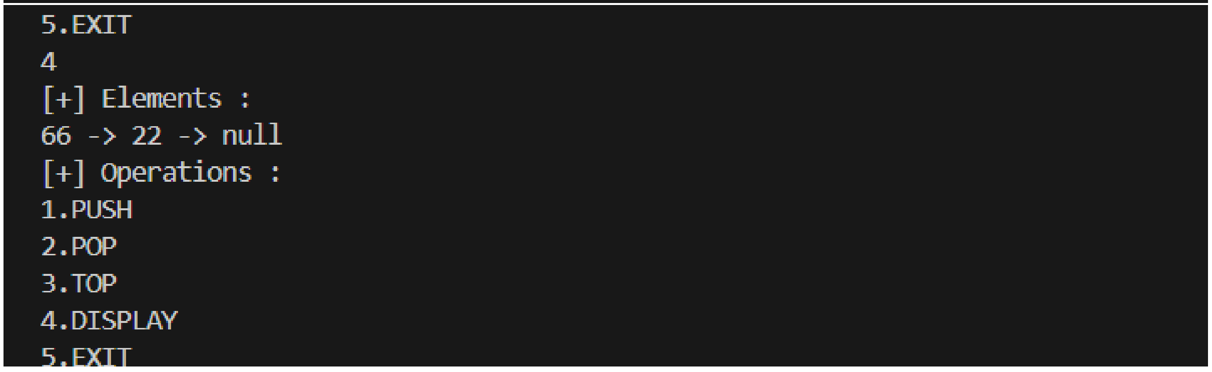
OUTPUT:



```
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
5.EXIT
4.DISPLAY
4.DISPLAY
5.EXIT
1
[-] Enter Element to Push
22
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
5.EXIT
1
[-] Enter Element to Push
4
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
5.EXIT
4
[+] Elements :
4 -> 22 -> null
```



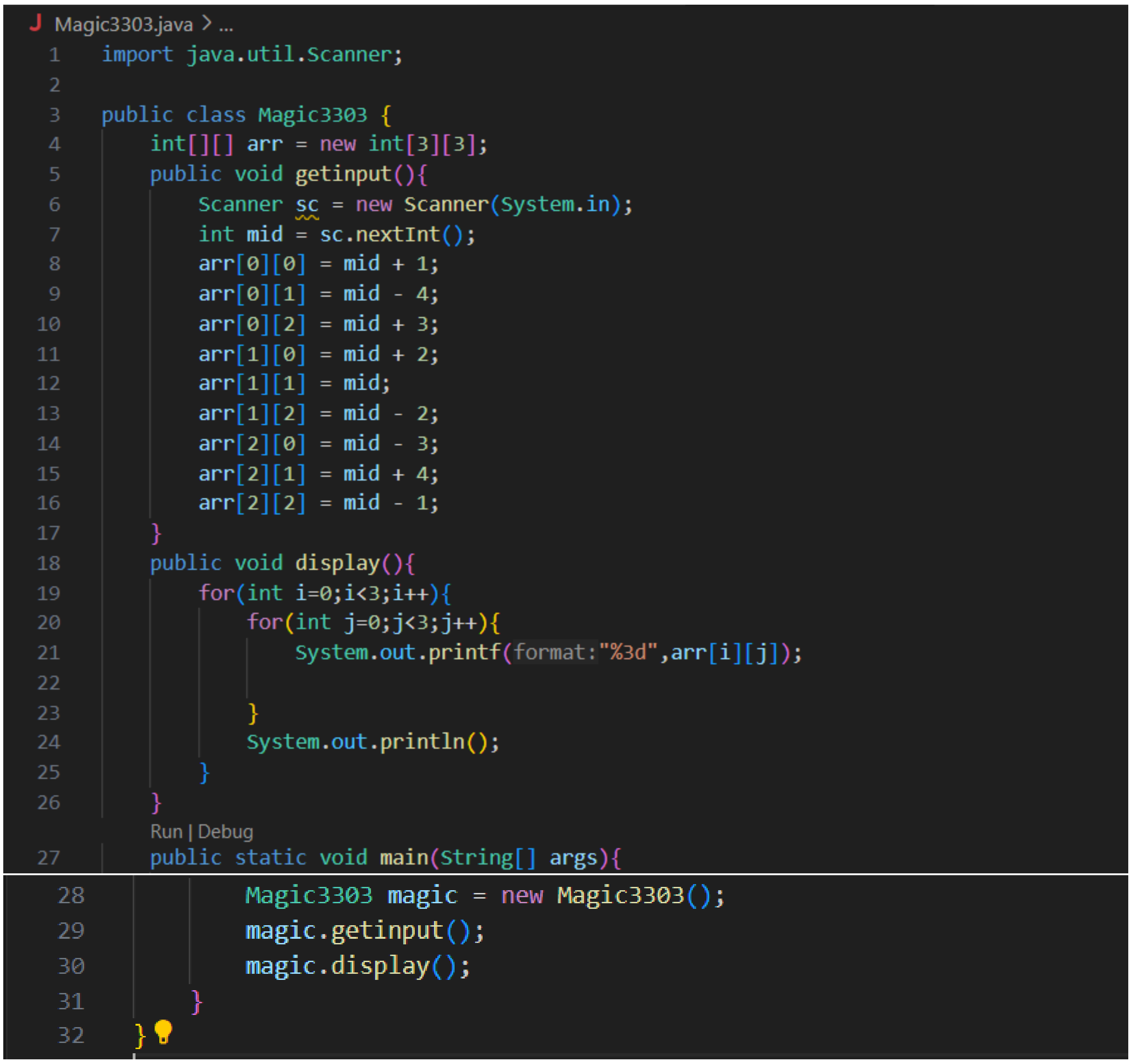
```
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
5.EXIT
2
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
5.EXIT
4
[+] Elements :
22 -> null
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
5.EXIT
1
[-] Enter Element to Push
66
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
```



```
5.EXIT
4
[+] Elements :
66 -> 22 -> null
[+] Operations :
1.PUSH
2.POP
3.TOP
4.DISPLAY
5.EXIT
```

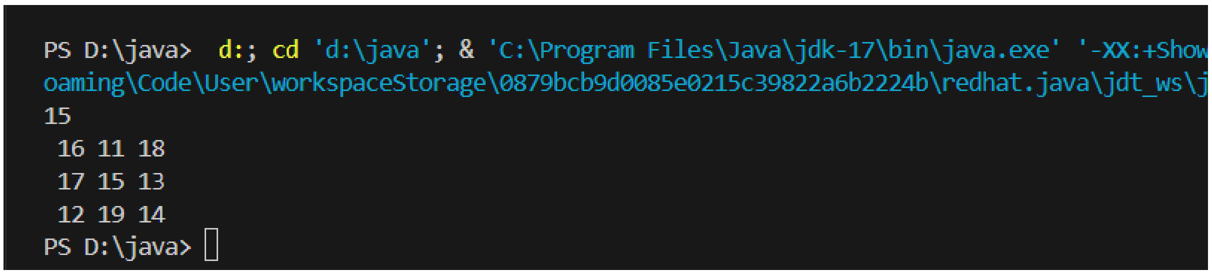
QUESTION 3 (ASSIGNEMENT – 2)Q4

MAGIC SQUARE:



```
Magic3303.java > ...
import java.util.Scanner;

public class Magic3303 {
    int[][] arr = new int[3][3];
    public void getinput(){
        Scanner sc = new Scanner(System.in);
        int mid = sc.nextInt();
        arr[0][0] = mid + 1;
        arr[0][1] = mid - 4;
        arr[0][2] = mid + 3;
        arr[1][0] = mid + 2;
        arr[1][1] = mid;
        arr[1][2] = mid - 2;
        arr[2][0] = mid - 3;
        arr[2][1] = mid + 4;
        arr[2][2] = mid - 1;
    }
    public void display(){
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                System.out.printf(format:"%3d",arr[i][j]);

            }
            System.out.println();
        }
    }
    Run | Debug
    public static void main(String[] args){
        Magic3303 magic = new Magic3303();
        magic.getinput();
        magic.display();
    }
}
```

OUTPUT:



```
PS D:\java>  d:; cd 'd:\java'; & 'C:\Program Files\Java\jdk-17\bin\java.exe' '-XX:+Show
oaming\Code\User\workspaceStorage\0879bcb9d0085e0215c39822a6b2224b\redhat.java\jdt_ws\j
15
 16 11 18
 17 15 13
 12 19 14
PS D:\java> 
```