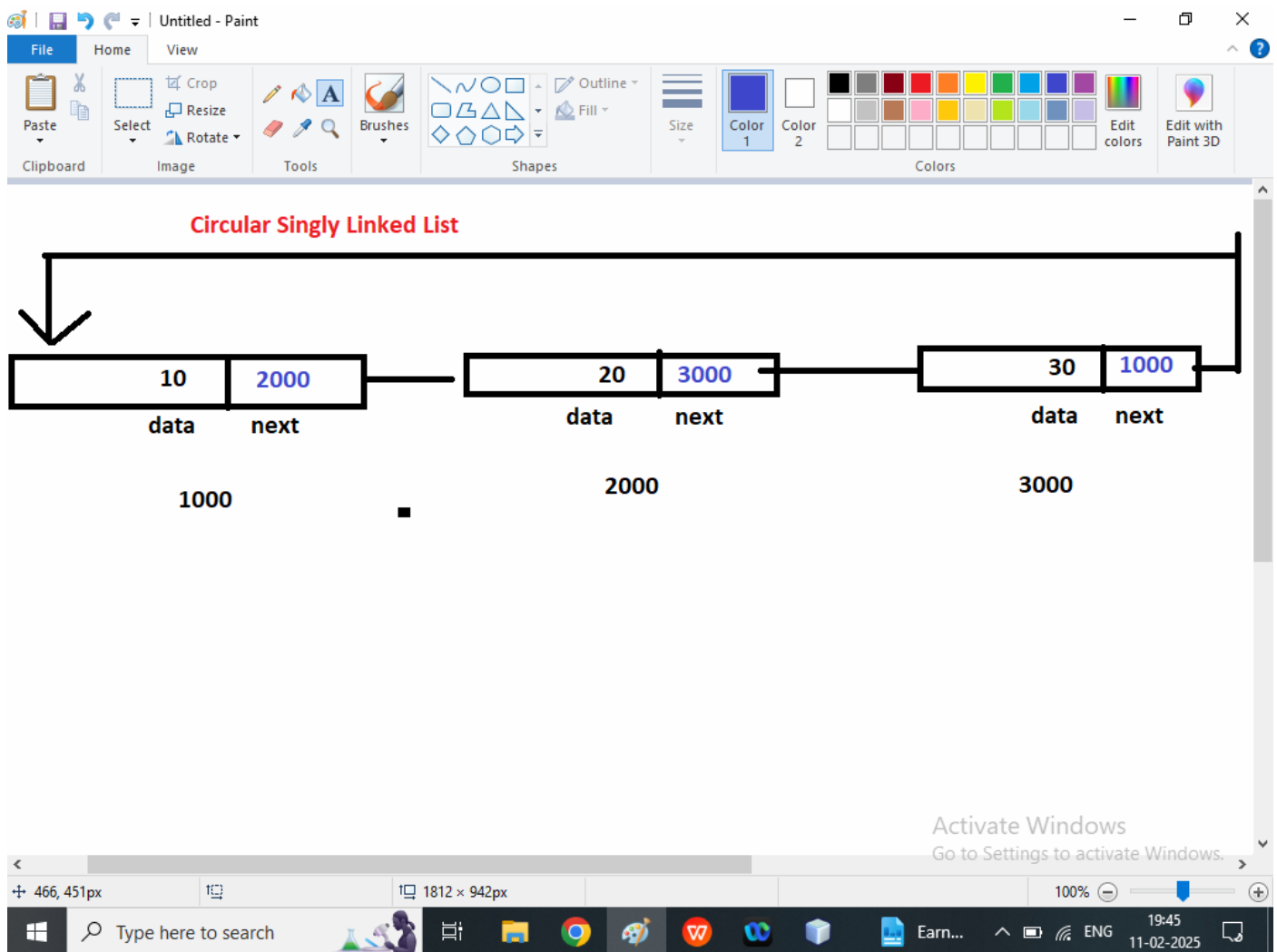


Q1. Example of Circular Singly Linked List and Circular doubly Linked List?

Ans:



```
package dsaj6;
```

```
public class Node {
```

```
    public int data;  
    public Node next;
```

```
    public Node(int data) {
```

```
        this.data = data;  
        next = null;  
        System.out.println("Node Created Success");  
    }
```

```
    public void display(Node head) {
```

```
        Node temp = head;  
        do {  
            System.out.print("==>" + temp.data);
```

```

temp = temp.next;
} while (temp != head);

```

```

}

```

```

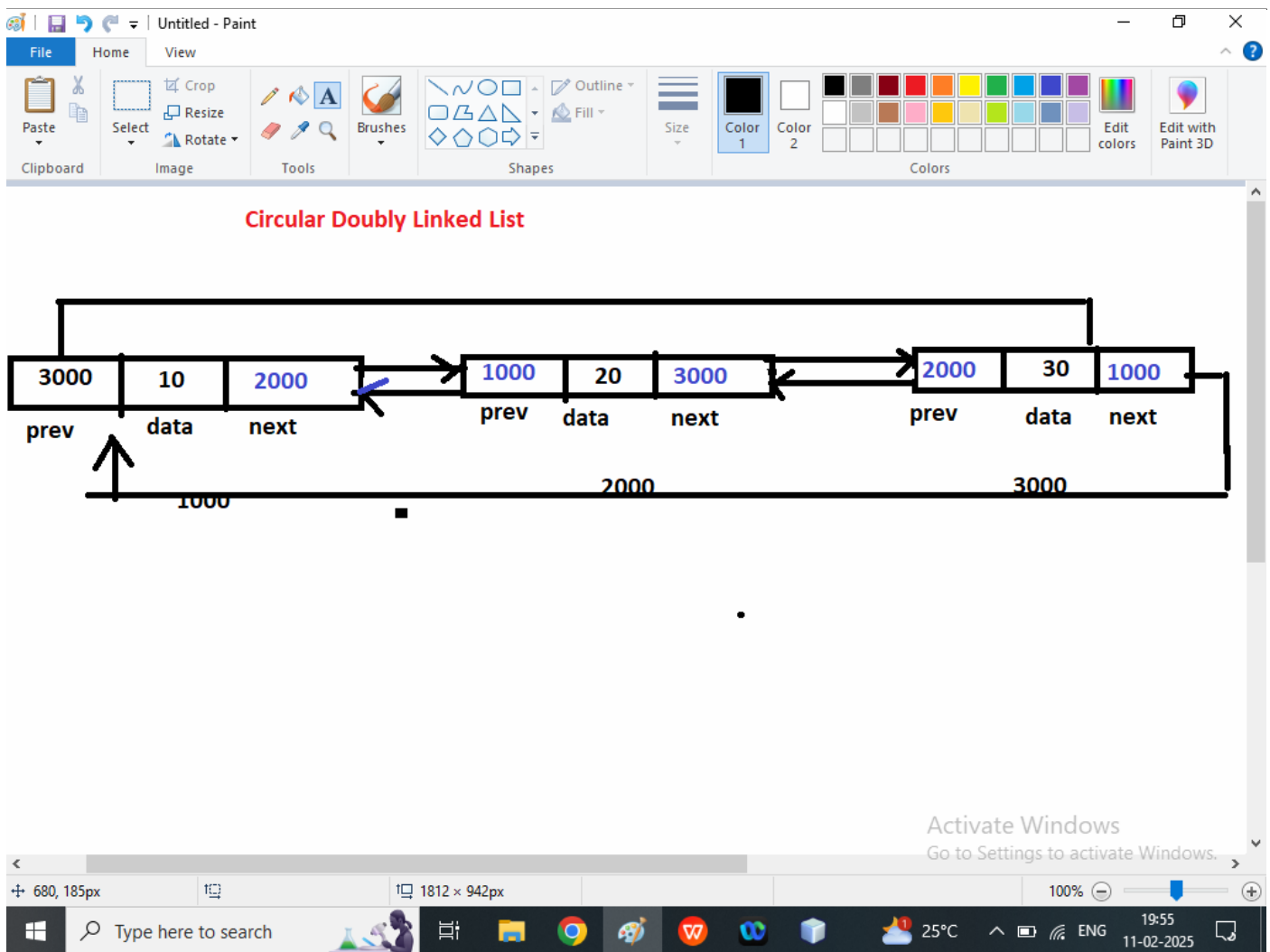
public static void main(String[] args) {
    Node f1 = new Node(10);
    Node f2 = new Node(20);
    Node f3 = new Node(30);
    f1.next = f2;
    f2.next = f3;
    f3.next = f1;
    Node head = f1;

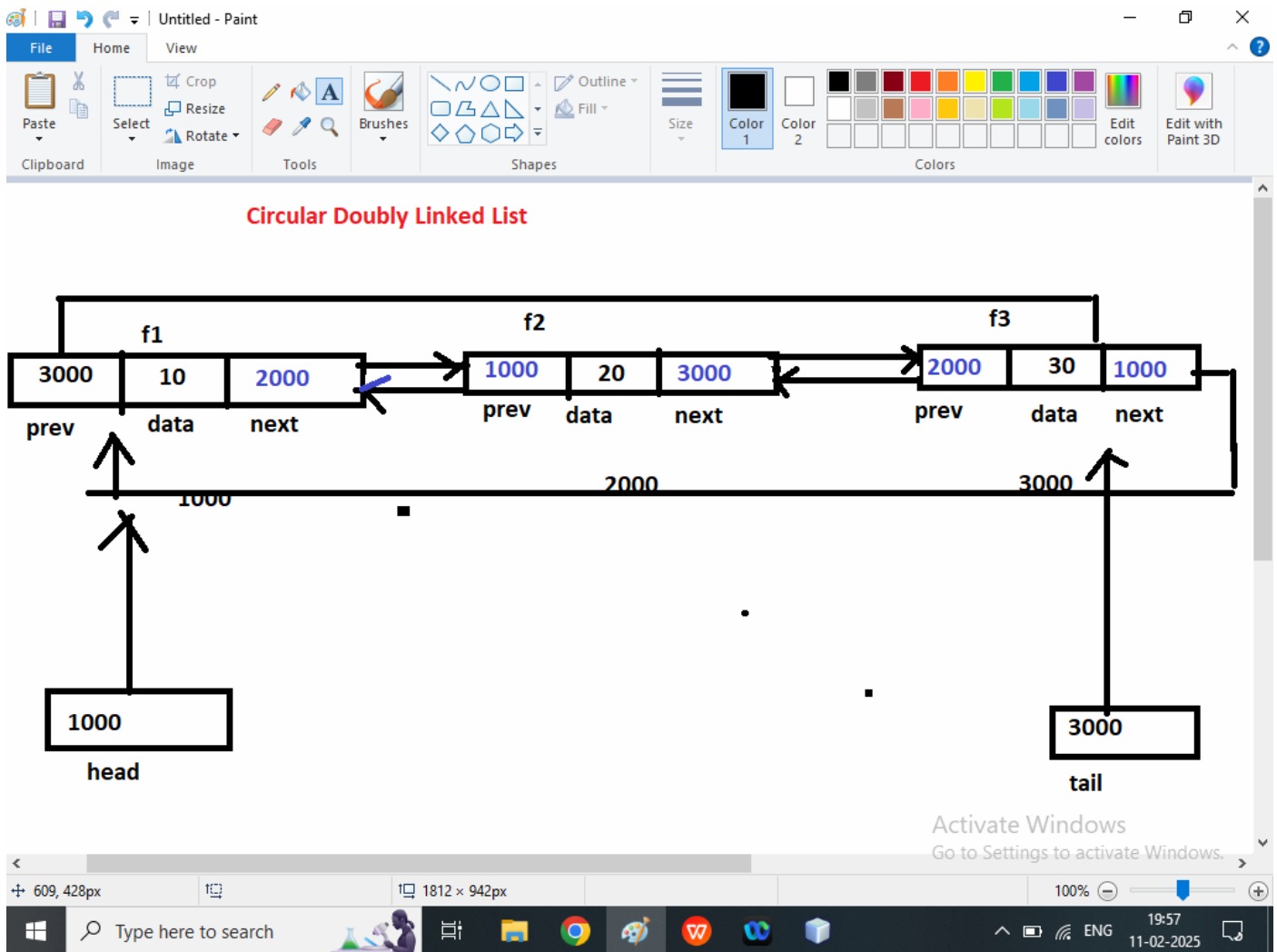
```

```

    System.out.println("Print Data of Singly Linked List");
    head.display(f3);
}

```





```
package dsaj6;
```

```
public class Node {
    public Node prev;
    public int data;
    public Node next;

    public Node(int data) {
        prev=null;
        this.data = data;
        next = null;
        System.out.println("Node Created Success");
    }
}
```

```
public void printForward(Node head) {
    Node temp = head;
    do {
        System.out.print("==>" + temp.data);
        temp = temp.next;
    } while (temp != head);
}
```

```
public void printBackward(Node head) {
    Node temp = head;
    do {
        System.out.print("==>" + temp.data);
        temp = temp.prev;
    } while (temp != head);
}
```

```

}

public static void main(String[] args) {
    Node f1 = new Node(10);
    Node f2 = new Node(20);
    Node f3 = new Node(30);

    f1.next=f2;
    f2.prev=f1;
    f2.next=f3;
    f3.prev=f2;
    f3.next=f1;
    f1.prev=f3;
    Node head = f1;
    Node tail=f3;
    System.out.println("Print Data of Singly Linked List forward direction ");
    head.printForward(f1);
    System.out.println("Print Data of Doubly Linked List in Backward Direction");
    head.printBackward(head);
}
}

```

Q1. Explain Bubble sort in data structure?

Ans: Bubble sort is a simple sorting algorithm that repeatedly swaps its adjacent elements if they are in the wrong order . This process is repeated until the array is sorted

Algorithm of Bubble sort

Step1: Compare Adjacent elements and swap them if needed

Step2: Move to the Next pair and repeat the above step

Step3: Continue this process until the entire array is sorted

Step4: After each pass, the largest element bubbles in its correct position

/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

package dsaj6;

/**

*

* @author Admin

*/

public class BSortNewClass {

public static void bubbleSort(int arr[]){

for(int i=0;i<arr.length;i++){//outer Loop ,0

for(int j=i+1;j<arr.length;j++){//inner Loop

if(arr[i]>arr[j]){

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

// System.out.println("Swapping Performed ");

}

}

// System.out.println("Number of Pass "+i);

}

}

public static void main(String[] args) {

int arr[]={5,3,8,4,6};

System.out.println("Print Data Before Sorting : \n ");

```
for(int x:arr){
    System.out.print("\t"+x);
}
bubbleSort(arr);
    System.out.println("\nPrint Data After Sorting : \n
");
for(int x:arr){
    System.out.print("\t"+x);
}
}
```

Time Complexity: $O(n^2)$

Space Complexity: $O(1)$