Folder AD_1

39 printable files

AD_1\Lab_10\StackUsingArray.java
AD_1\Lab_10\StackUsingLinkedList.java
AD_1\Lab_11\QueueUsingArray.java
AD_1\Lab_11\QueueUsingLinkedList.java
AD_1\Lab_1\FindMaxMin.java
AD_1\Lab_1\RotateArray.java
AD_1\Lab_1\RotateArray_1.java
AD_1\Lab_1\SumOfNNumbers.java
AD_1\Lab_2\Factorial.java
AD_1\Lab_2\LargestSumSubarray.java
AD_1\Lab_2\MaxMinArray.java
AD_1\Lab_2\NthFibonacci.java
AD_1\Lab_2\SmallestPositiveMissingNumber.java
AD_1\Lab_3\DecimalToHexadecimal.java
AD_1\Lab_3\RecursiveFactorial.java
AD_1\Lab_3\RecursiveFibonacci.java
AD_1\Lab_3\RecursiveGCD.java
AD_1\Lab_3\RecursiveMaxMin.java
AD_1\Lab_3\RecursivePower.java
AD_1\Lab_3\RecursiveSmallestPositiveMissingNumber.java
AD_1\Lab_3\RecursiveSum.java
AD_1\Lab_4\BubbleSort.java
AD_1\Lab_4\InsertionSort.java
AD_1\Lab_4\SelectionSort.java
AD_1\Lab_5\ArrayReduction.java
AD_1\Lab_5\CheckReverse.java
AD_1\Lab_5\MergeSortedArrays.java
AD_1\Lab_5\MergingArrays.java
AD_1\Lab_6\BinarySearch.java
AD_1\Lab_6\LinearSearch.java
AD_1\Lab_6\RecursiveBinarySearch.java
AD_1\Lab_6\RecursiveLinearSearch.java
AD_1\Lab_7\FirstRepeatedElement.java
AD_1\Lab_7\MaxFrequencyElement.java
AD_1\Lab_7\MinMaxDifference.java
AD_1\Lab_7\MissingNumber.java
AD_1\Lab_7\PrintDuplicates.java
AD_1\Lab_8\LinkedListOperations.java
AD_1\Lab_9\Javacollection.java

AD_1\Lab_10\StackUsingArray.java

```java
package AD_1.Lab_10;

import java.util.Scanner;
public class StackUsingArray
{
        public static boolean isEmpty(int top){
            return (top==-1);
        }
        public static boolean isFull(int top){
            return (top==MAX-1);
        }
        ////////////////////////////////////////////////////////////
        public static int push(int S[],int top)
        {
            if(isFull(top))
            {
                System.out.println("Stack is full(Overflow)");
```

```
    }
    else{
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter data to push: ");
        int x=sc.nextInt();
        S[++top]=x;
        sc.close();
    }
    return top;

}
////////////////////////////////////////////////////////////
public static int pop(int S[],int top){
    if(isEmpty(top)){
        System.out.println("Stack is empty(Underflow)");
    }
    else{
        System.out.println("Popped element is: "+S[top]);
        top--;
    }
    return top;
}
////////////////////////////////////////////////////////////
public static void display(int S[],int top){
    if(isEmpty(top)) System.out.println("Nothing to display,Stack is empty!!");
    else{
        System.out.println("The stack elements are:::::::::");
        for(int i=0;i<=top;i++)
            System.out.print(S[i]+" ");
        System.out.println();
    }
}
////////////////////////////////////////////////////////////

public static final int MAX=10;

public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    int stack[]=new int[MAX];
    int top=-1;
    while(true)
    {
        System.out.println("***MENU***");
        System.out.println("0: Exit");
        System.out.println("1: Push");
        System.out.println("2: Pop");
        System.out.println("3: Display");
        System.out.println("Enter your choice");
        int choice=sc.nextInt();
        switch(choice)
        {
            case 0:
                System.exit(0);
            case 1:
                top=push(stack,top);
                break;
```

```
                    case 2:
                        top=pop(stack,top);
                        break;
                    case 3:
                        display(stack,top);
                        break;
                    default:
                        System.out.println("Invalid choice");
                }//switch
                sc.close();
            }//while
        }//main
    }//class


/*

    |-----------------------------------|
    |    ::::       OUTPUT      ::::    |
    |-----------------------------------|


***MENU***
0: Exit
1: Push
2: Pop
3: Display
Enter your choice
1
Enter data to push:
25
***MENU***
0: Exit
1: Push
2: Pop
3: Display
Enter your choice
1
Enter data to push:
67
***MENU***
0: Exit
1: Push
2: Pop
3: Display
Enter your choice
1
Enter data to push:
55
***MENU***
0: Exit
1: Push
2: Pop
3: Display
Enter your choice
3
```

```
The stack elements are:::::::::
25 67 55
***MENU***
0: Exit
1: Push
2: Pop
3: Display
Enter your choice
2
Popped element is: 55
***MENU***
0: Exit
1: Push
2: Pop
3: Display
Enter your choice
3
The stack elements are:::::::::
25 67
 */
```

AD_1\Lab_10\StackUsingLinkedList.java

```java
package AD_1.Lab_10;

public class StackUsingLinkedList {
    static class Node {
        int data;
        Node next;

        public Node(int data) {
            this.data = data;
            this.next = null;
        }
    }

    static class Stack {
        Node top;

        public Stack() {
            top = null;
        }

        // Push an element to the stack
        public void push(int data) {
            Node newNode = new Node(data);
            newNode.next = top;
            top = newNode;
            System.out.println(data + " pushed to stack");
        }

        // Pop an element from the stack
        public int pop() {
            if (top == null) {
                System.out.println("Stack is empty");
                return -1;
```

```
        }
        int popped = top.data;
        top = top.next;
        return popped;
    }

    // Peek the top element of the stack
    public int peek() {
        if (top == null) {
            System.out.println("Stack is empty");
            return -1;
        }
        return top.data;
    }

    // Check if the stack is empty
    public boolean isEmpty() {
        return top == null;
    }

    // Print the stack
    public void printStack() {
        Node temp = top;
        if (temp == null) {
            System.out.println("Stack is empty");
            return;
        }
        System.out.print("Stack: ");
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.next;
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    Stack stack = new Stack();

    // Performing stack operations
    stack.push(10);
    stack.push(20);
    stack.push(30);
    stack.push(40);

    stack.printStack(); // Output: Stack: 40 30 20 10

    System.out.println("Top element is " + stack.peek()); // Output: Top element is 40

    System.out.println(stack.pop() + " popped from stack"); // Output: 40 popped from stack

    stack.printStack(); // Output: Stack: 30 20 10

    System.out.println("Is stack empty? " + stack.isEmpty()); // Output: Is stack empty? false
}
}
```

AD_1\Lab_11\QueueUsingArray.java

```java
package AD_1.Lab_11;


import java.util.Scanner;
public class QueueUsingArray {

    public static void insert(int Q[])//adding an element x to the rear end of the queue Q
    {
        if(is_full()) System.out.println("Queue is full(Overflow)");
        else{
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter data to insert: ");
            int x=sc.nextInt();
            Q[++rear]=x;
            sc.close();
        }
        if(rear==0) front=0;
    }
    public static void delete(int Q[])//deletes the element from the front of the queue Q
    {
        if(is_empty()) System.out.println("Queue is empty(Underflow)");
        else{
            System.out.println("Deleted element is: "+Q[front]);
            if(front==rear) {
                front=-1;
                rear=-1;
            }
            else
                front++;
        }
    }
    public static void display(int Q[])//display all the elements of the queue Q.
    {
        if(is_empty()) System.out.println("Queue is empty");
        else{
            System.out.println("Elements of queue are:::: ");
            for(int i=front;i<=rear;i++)
                System.out.print(Q[i]+" ");
            System.out.println();
        }
    }
    public static boolean is_full()//check if the queue is full or not.
    {
        return (rear==MAX-1);
    }
    public static boolean is_empty()//check if the queue is empty or not.
    {
        return (rear==-1 || front==-1);
    }

    public static final int MAX=5;
    public static int front=-1;
```

```
    public static int rear=-1;

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);
        int queue[]=new int[MAX];

        while(true)
        {

            System.out.println("***MENU***");
            System.out.println("0: Exit");
            System.out.println("1: Insert");
            System.out.println("2: Delete");
            System.out.println("3: Display");
            System.out.println("Enter your choice");
            int choice=sc.nextInt();
            switch(choice)
            {
                case 0:
                    System.exit(0);
                case 1:
                    insert(queue);
                    break;
                case 2:
                    delete(queue);
                    break;
                case 3:
                    display(queue);
                    break;
                default:
                    System.out.println("Invalid choice");
            }//switch
            sc.close();
        }//while
    }//main
}//class


/*

    |-----------------------------------|
    |    ::::       OUTPUT      ::::    |
    |-----------------------------------|


***MENU***
0: Exit
1: Insert
2: Delete
3: Display
Enter your choice
1
Enter data to insert:
25
***MENU***
```

```
0: Exit
1: Insert
2: Delete
3: Display
Enter your choice
1
Enter data to insert:
45
***MENU***
0: Exit
1: Insert
2: Delete
3: Display
Enter your choice
1
Enter data to insert:
65
***MENU***
0: Exit
1: Insert
2: Delete
3: Display
Enter your choice
3
Elements of queue are::::
25 45 65
***MENU***
0: Exit
1: Insert
2: Delete
3: Display
Enter your choice
2
Deleted element is: 25
***MENU***
0: Exit
1: Insert
2: Delete
3: Display
Enter your choice
3
Elements of queue are::::
45 65
***MENU***
0: Exit
1: Insert
2: Delete
3: Display
Enter your choice
0

Process finished with exit code 0

 */
```

AD_1\Lab_11\QueueUsingLinkedList.java

```java
package AD_1.Lab_11;

public class QueueUsingLinkedList {
    static class Node {
        int data;
        Node next;

        public Node(int data) {
            this.data = data;
            this.next = null;
        }
    }

    static class Queue {
        Node front, rear;

        public Queue() {
            front = rear = null;
        }

        // Enqueue (add element to the queue)
        public void enqueue(int data) {
            Node newNode = new Node(data);
            if (rear == null) {
                front = rear = newNode;
                return;
            }
            rear.next = newNode;
            rear = newNode;
            System.out.println(data + " enqueued to queue");
        }

        // Dequeue (remove element from the queue)
        public int dequeue() {
            if (front == null) {
                System.out.println("Queue is empty");
                return -1;
            }
            int dequeued = front.data;
            front = front.next;
            if (front == null) {
                rear = null;
            }
            return dequeued;
        }

        // Peek the front element of the queue
        public int peek() {
            if (front == null) {
                System.out.println("Queue is empty");
                return -1;
            }
            return front.data;
        }

        // Check if the queue is empty
```

```
        public boolean isEmpty() {
            return front == null;
        }

        // Print the queue
        public void printQueue() {
            Node temp = front;
            if (temp == null) {
                System.out.println("Queue is empty");
                return;
            }
            System.out.print("Queue: ");
            while (temp != null) {
                System.out.print(temp.data + " ");
                temp = temp.next;
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Queue queue = new Queue();

        // Performing queue operations
        queue.enqueue(10);
        queue.enqueue(20);
        queue.enqueue(30);
        queue.enqueue(40);

        queue.printQueue(); // Output: Queue: 10 20 30 40

        System.out.println("Front element is " + queue.peek()); // Output: Front element is 10

        System.out.println(queue.dequeue() + " dequeued from queue"); // Output: 10 dequeued from queue

        queue.printQueue(); // Output: Queue: 20 30 40

        System.out.println("Is queue empty? " + queue.isEmpty()); // Output: Is queue empty? false
    }
}
```

AD_1\Lab_1\FindMaxMin.java

```
package AD_1.Lab_1;

import java.util.Scanner;

public class FindMaxMin {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] numbers = new int[n];

        System.out.println("Enter the numbers:");
```

```
        for (int i = 0; i < n; i++) {
            numbers[i] = scanner.nextInt();
        }

        int max = numbers[0], min = numbers[0];
        for (int i = 1; i < n; i++) {
            if (numbers[i] > max) {
                max = numbers[i];
            }
            if (numbers[i] < min) {
                min = numbers[i];
            }
        }

        System.out.println("Maximum: " + max);
        System.out.println("Minimum: " + min);
        scanner.close();
    }
}
```

AD_1\Lab_1\RotateArray.java

```
package AD_1.Lab_1;

import java.util.Scanner;

public class RotateArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the value of k: ");
        int k = scanner.nextInt();
        k = k % n;

        System.out.println("Array after rotation:");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[(i + k) % n] + " ");
        }
        scanner.close();
    }
}
```

AD_1\Lab_1\RotateArray_1.java

```
package AD_1.Lab_1;
```

```java
import java.util.Scanner;

public class RotateArray_1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.print("Enter the number of positions to rotate: ");
        int k = sc.nextInt();
        rotateArray(arr, k, n);
        System.out.print("Rotated Array: ");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        sc.close();
    }

    public static void rotateArray(int[] arr, int k, int n) {
        k = k % n;
        reverseArray(arr, 0, n - 1);
        reverseArray(arr, 0, k - 1);
        reverseArray(arr, k, n - 1);
    }

    public static void reverseArray(int[] arr, int start, int end) {
        while (start < end) {
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }
    }
}
```

AD_1\Lab_1\SumOfNNumbers.java

```java
package AD_1.Lab_1;

import java.util.Scanner;

public class SumOfNNumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] numbers = new int[n];
        int sum = 0;
```

```
        System.out.println("Enter the numbers:");
        for (int i = 0; i < n; i++) {
            numbers[i] = scanner.nextInt();
            sum += numbers[i];
        }

        System.out.println("Sum of numbers: " + sum);
        scanner.close();
    }
}
```

AD_1\Lab_2\Factorial.java

```
package AD_1.Lab_2;

import java.util.Scanner;

public class Factorial {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = scanner.nextInt();

        long factorial = 1;
        for (int i = 1; i <= n; i++) {
            factorial *= i;
        }

        System.out.println("Factorial of " + n + ": " + factorial);
        scanner.close();
    }
}
```

AD_1\Lab_2\LargestSumSubarray.java

```
package AD_1.Lab_2;

import java.util.Scanner;

public class LargestSumSubarray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
```

```
        int maxSum = Integer.MIN_VALUE, currentSum = 0;
        for (int i = 0; i < n; i++) {
            currentSum += arr[i];
            if (currentSum > maxSum) {
                maxSum = currentSum;
            }
            if (currentSum < 0) {
                currentSum = 0;
            }
        }

        System.out.println("Largest sum of contiguous subarray: " + maxSum);
        scanner.close();
    }
}
```

AD_1\Lab_2\MaxMinArray.java

```
package AD_1.Lab_2;

import java.util.Arrays;
import java.util.Scanner;

public class MaxMinArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
        Arrays.sort(arr);

        int[] result = new int[n];
        int start = 0, end = n - 1;
        boolean flag = true;

        for (int i = 0; i < n; i++) {
            if (flag) {
                result[i] = arr[end--];
            } else {
                result[i] = arr[start++];
            }
            flag = !flag;
        }

        System.out.println("Maximum minimum array:");
        for (int num : result) {
            System.out.print(num + " ");
        }
        scanner.close();
    }
```

```
}
```

AD_1\Lab_2\NthFibonacci.java

```
package AD_1.Lab_2;
import java.util.Scanner;

public class NthFibonacci {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of n: ");
        int n = sc.nextInt();
        System.out.println("Fibonacci number at position " + n + " is: " + fibonacci(n));
        sc.close();
    }

    public static int fibonacci(int n) {
        if (n <= 1) return n;
        int a = 0, b = 1, fib = 0;
        for (int i = 2; i <= n; i++) {
            fib = a + b;
            a = b;
            b = fib;
        }
        return fib;
    }
}
```

AD_1\Lab_2\SmallestPositiveMissingNumber.java

```
package AD_1.Lab_2;

public class SmallestPositiveMissingNumber {
    public static void main(String[] args) {
        int[] arr = {3, 4, -1, 1, 5, 2};
        int missingNumber = findSmallestMissingNumber(arr);
        System.out.println("Smallest positive missing number: " + missingNumber);
    }

    public static int findSmallestMissingNumber(int[] arr) {
        int n = arr.length;


        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    // Swap arr[j] and arr[j + 1]
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
```

```
        int smallestMissing = 1;
        for (int num : arr) {
            if (num == smallestMissing) {
                smallestMissing++;
            }
        }
        return smallestMissing;
    }
}
```

AD_1\Lab_3\DecimalToHexadecimal.java

```
package AD_1.Lab_3;

import java.util.Scanner;

public class DecimalToHexadecimal {
    public static String decimalToHex(int num) {
        if (num == 0) {
            return "";
        }
        int remainder = num % 16;
        char hexDigit = (char) (remainder < 10 ? '0' + remainder : 'A' + remainder - 10);
        return decimalToHex(num / 16) + hexDigit;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a decimal number: ");
        int num = scanner.nextInt();

        String hex = decimalToHex(num);
        System.out.println("Hexadecimal: " + (hex.isEmpty() ? "0" : hex));
        scanner.close();
    }
}
```

AD_1\Lab_3\RecursiveFactorial.java

```
package AD_1.Lab_3;

import java.util.Scanner;

public class RecursiveFactorial {
    public static long factorial(int n) {
        if (n <= 1) {
            return 1;
        }
        return n * factorial(n - 1);
    }

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

```
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = scanner.nextInt();

        System.out.println("Factorial of " + n + ": " + factorial(n));
        scanner.close();
    }
}
```

AD_1\Lab_3\RecursiveFibonacci.java

```
package AD_1.Lab_3;

import java.util.Scanner;

public class RecursiveFibonacci {
    public static int fibonacci(int n) {
        if (n == 0) {
            return 0;
        }
        if (n == 1) {
            return 1;
        }
        return fibonacci(n - 1) + fibonacci(n - 2);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the value of n: ");
        int n = scanner.nextInt();

        System.out.println("Fibonacci number: " + fibonacci(n));
        scanner.close();
    }
}
```

AD_1\Lab_3\RecursiveGCD.java

```
package AD_1.Lab_3;

import java.util.Scanner;

public class RecursiveGCD {
    public static int gcd(int a, int b) {
        if (b == 0) {
            return a;
        }
        return gcd(b, a % b);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter two numbers: ");
```

```
        int a = scanner.nextInt();
        int b = scanner.nextInt();

        System.out.println("GCD of " + a + " and " + b + ": " + gcd(a, b));
        scanner.close();
    }
}
```

AD_1\Lab_3\RecursiveMaxMin.java

```
package AD_1.Lab_3;

import java.util.Scanner;

public class RecursiveMaxMin {
    public static int findMax(int[] arr, int n) {
        if (n == 1) {
            return arr[0];
        }
        return Math.max(arr[n - 1], findMax(arr, n - 1));
    }

    public static int findMin(int[] arr, int n) {
        if (n == 1) {
            return arr[0];
        }
        return Math.min(arr[n - 1], findMin(arr, n - 1));
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the numbers:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.println("Maximum: " + findMax(arr, n));
        System.out.println("Minimum: " + findMin(arr, n));
        scanner.close();
    }
}
```

AD_1\Lab_3\RecursivePower.java

```
package AD_1.Lab_3;

import java.util.Scanner;
```

```java
public class RecursivePower {
    public static long power(int base, int exp) {
        if (exp == 0) {
            return 1;
        }
        return base * power(base, exp - 1);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the base: ");
        int base = scanner.nextInt();
        System.out.print("Enter the exponent: ");
        int exp = scanner.nextInt();

        System.out.println(base + " raised to the power " + exp + ": " + power(base, exp));
        scanner.close();
    }
}
```

AD_1\Lab_3\RecursiveSmallestPositiveMissingNumber.java

```java
package AD_1.Lab_3;

import java.util.Scanner;

public class RecursiveSmallestPositiveMissingNumber {
    public static int findMissing(int[] arr, int n, int num) {
        if (n == 0) return num;

        if (arr[n - 1] == num) {
            return findMissing(arr, arr.length, num + 1);
        }
        return findMissing(arr, n - 1, num);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of the array: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.println("Smallest positive missing number: " + findMissing(arr, n, 1));
        scanner.close();
    }
}
```

AD_1\Lab_3\RecursiveSum.java

```java
package AD_1.Lab_3;

import java.util.Scanner;

public class RecursiveSum {
    public static int sum(int[] arr, int n) {
        if (n <= 0) {
            return 0;
        }
        return arr[n - 1] + sum(arr, n - 1);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the numbers:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.println("Sum of numbers: " + sum(arr, n));
        scanner.close();
    }
}
```

AD_1\Lab_4\BubbleSort.java

```java
package AD_1.Lab_4;

import java.util.Scanner;

public class BubbleSort {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
```

```
        System.out.println("Sorted array:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
        scanner.close();
    }
}
```

AD_1\Lab_4\InsertionSort.java

```
package AD_1.Lab_4;

import java.util.Scanner;

public class InsertionSort {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        for (int i = 1; i < n; i++) {
            int key = arr[i];
            int j = i - 1;

            while (j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j--;
            }
            arr[j + 1] = key;
        }

        System.out.println("Sorted array:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
        scanner.close();
    }
}
```

AD_1\Lab_4\SelectionSort.java

```
package AD_1.Lab_4;

import java.util.Scanner;

public class SelectionSort {
```

```
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        for (int i = 0; i < n - 1; i++) {
            int minIndex = i;

            for (int j = i + 1; j < n; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }

            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }

        System.out.println("Sorted array:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
        scanner.close();
    }
}
```

AD_1\Lab_5\ArrayReduction.java

```
package AD_1.Lab_5;

import java.util.Scanner;

public class ArrayReduction {
    public static void reduceArray(int[] arr) {
        int count = 0;
        while (true) {
            int min = findSmallestNonZero(arr);
            if (min == 0) break;

            for (int i = 0; i < arr.length; i++) {
                if (arr[i] > 0) arr[i] -= min;
            }

            for (int i = 0; i < arr.length; i++)
                System.out.print(arr[i] + " ");
            System.out.println();

            count++;
```

```
        }
        System.out.println("Number of Reduction steps = " + count);
    }

    public static int findSmallestNonZero(int[] arr) {
        int min = Integer.MAX_VALUE;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] > 0 && arr[i] < min) min = arr[i];
        }
        return (min == Integer.MAX_VALUE) ? 0 : min;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the size of array: ");
        int n = sc.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter array elements:");
        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt();

        reduceArray(arr);
        sc.close();
    }
}


// Enter the size of array: 5
// Enter array elements:
// 2
// 4
// 9
// 8
// 15
// 0 2 7 6 13
// 0 0 5 4 11
// 0 0 1 0 7
// 0 0 0 0 6
// 0 0 0 0 0
```

AD_1\Lab_5\CheckReverse.java

```
package AD_1.Lab_5;

import java.util.Scanner;

public class CheckReverse {
    public static boolean canReverseToSort(int[] arr) {
        int start = -1, end = -1;

        for (int i = 0; i < arr.length - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                if (start == -1) {
                    start = i;
                }
```

```
                end = i + 1;
            }
        }

        if (start == -1) {
            return true; // Already sorted
        }

        while (start < end) {
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }

        for (int i = 0; i < arr.length - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                return false;
            }
        }

        return true;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        if (canReverseToSort(arr)) {
            System.out.println("Array can be sorted by reversing a single subarray.");
        } else {
            System.out.println("Array cannot be sorted by reversing a single subarray.");
        }
        scanner.close();
    }
}
```

AD_1\Lab_5\MergeSortedArrays.java

```
package AD_1.Lab_5;

import java.util.Scanner;

public class MergeSortedArrays {
    public static void mergeArrays(int[] arr1, int[] arr2, int[] merged) {
        int i = 0, j = 0, k = 0;
```

```
        while (i < arr1.length && j < arr2.length) {
            if (arr1[i] <= arr2[j]) {
                merged[k++] = arr1[i++];
            } else {
                merged[k++] = arr2[j++];
            }
        }

        while (i < arr1.length) {
            merged[k++] = arr1[i++];
        }

        while (j < arr2.length) {
            merged[k++] = arr2[j++];
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the size of the first sorted array: ");
        int n1 = scanner.nextInt();
        int[] arr1 = new int[n1];
        System.out.println("Enter elements of the first array:");
        for (int i = 0; i < n1; i++) {
            arr1[i] = scanner.nextInt();
        }

        System.out.print("Enter the size of the second sorted array: ");
        int n2 = scanner.nextInt();
        int[] arr2 = new int[n2];
        System.out.println("Enter elements of the second array:");
        for (int i = 0; i < n2; i++) {
            arr2[i] = scanner.nextInt();
        }

        int[] merged = new int[n1 + n2];
        mergeArrays(arr1, arr2, merged);

        System.out.println("Merged array:");
        for (int num : merged) {
            System.out.print(num + " ");
        }
        scanner.close();
    }
}
```

AD_1\Lab_5\MergingArrays.java

```
package AD_1.Lab_5;

public class MergingArrays {
    /*
      Input:
      arr1 = {1, 5, 9, 10, 15, 20}
```

```
  arr2 = {2, 3, 8, 13}

  Output:
  arr1 = {1, 2, 3, 5, 8, 9}
  arr2 = {10, 13, 15, 20}
  Merged Array: {1, 2, 3, 5, 8, 9, 10, 13, 15, 20}
*/

public static void mergeSortedArrays(int[] arr1, int size1, int[] arr2, int size2) {
    int index = 0;

    while (index < size1) {
        if (arr1[index] <= arr2[0]) {
            index++;
        } else {
            // Swap the element from arr1 with the smallest element of arr2
            int temp = arr1[index];
            arr1[index] = arr2[0];
            arr2[0] = temp;
            index++;

            // Reorganize arr2 to maintain its sorted order
            for (int i = 0; i < size2 - 1; i++) {
                if (arr2[i] < arr2[i + 1]) {
                    break;
                }
                int temp2 = arr2[i];
                arr2[i] = arr2[i + 1];
                arr2[i + 1] = temp2;
            }
        }
    }
}

public static void main(String[] args) {
    int[] arr1 = {1, 5, 9, 10, 15, 20};
    int[] arr2 = {2, 3, 8, 13};

    System.out.println("Initial Arrays:");
    System.out.print("arr1: ");
    for (int num : arr1)
        System.out.print(num + " ");
    System.out.println();

    System.out.print("arr2: ");
    for (int num : arr2)
        System.out.print(num + " ");
    System.out.println();

    mergeSortedArrays(arr1, arr1.length, arr2, arr2.length);

    System.out.println("\nFinal Arrays:");
    System.out.print("arr1: ");
    for (int num : arr1)
        System.out.print(num + " ");
    System.out.println();
```

```
        System.out.print("arr2: ");
        for (int num : arr2)
            System.out.print(num + " ");
        System.out.println();

        // Combine arr1 and arr2 into a merged array
        int[] mergedArray = new int[arr1.length + arr2.length];
        System.arraycopy(arr1, 0, mergedArray, 0, arr1.length);
        System.arraycopy(arr2, 0, mergedArray, arr1.length, arr2.length);

        System.out.println("\nMerged Array:");
        for (int num : mergedArray)
            System.out.print(num + " ");
    }
}
```

AD_1\Lab_6\BinarySearch.java

```
package AD_1.Lab_6;

import java.util.Scanner;

public class BinarySearch {
    public static int binarySearch(int[] arr, int target) {
        int low = 0, high = arr.length - 1;

        while (low <= high) {
            int mid = low + (high - low) / 2;

            if (arr[mid] == target) {
                return mid;
            } else if (arr[mid] < target) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }

        return -1;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements in sorted order:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the target value to search: ");
        int target = scanner.nextInt();
```

```
        int result = binarySearch(arr, target);
        if (result != -1) {
            System.out.println("Element found at index: " + result);
        } else {
            System.out.println("Element not found.");
        }
        scanner.close();
    }
}
```

AD_1\Lab_6\LinearSearch.java

```
package AD_1.Lab_6;

import java.util.Scanner;

public class LinearSearch {
    public static int linearSearch(int[] arr, int target) {
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == target) {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the target value to search: ");
        int target = scanner.nextInt();

        int result = linearSearch(arr, target);
        if (result != -1) {
            System.out.println("Element found at index: " + result);
        } else {
            System.out.println("Element not found.");
        }
        scanner.close();
    }
}
```

AD_1\Lab_6\RecursiveBinarySearch.java

```java
package AD_1.Lab_6;

import java.util.Scanner;

public class RecursiveBinarySearch {
    public static int binarySearch(int[] arr, int target, int low, int high) {
        if (low > high) {
            return -1;
        }

        int mid = low + (high - low) / 2;

        if (arr[mid] == target) {
            return mid;
        } else if (arr[mid] < target) {
            return binarySearch(arr, target, mid + 1, high);
        } else {
            return binarySearch(arr, target, low, mid - 1);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements in sorted order:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the target value to search: ");
        int target = scanner.nextInt();

        int result = binarySearch(arr, target, 0, n - 1);
        if (result != -1) {
            System.out.println("Element found at index: " + result);
        } else {
            System.out.println("Element not found.");
        }
        scanner.close();
    }
}
```

AD_1\Lab_6\RecursiveLinearSearch.java

```java
package AD_1.Lab_6;

import java.util.Scanner;

public class RecursiveLinearSearch {
```

```
    public static int linearSearch(int[] arr, int target, int index) {
        if (index >= arr.length) {
            return -1;
        }
        if (arr[index] == target) {
            return index;
        }
        return linearSearch(arr, target, index + 1);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the target value to search: ");
        int target = scanner.nextInt();

        int result = linearSearch(arr, target, 0);
        if (result != -1) {
            System.out.println("Element found at index: " + result);
        } else {
            System.out.println("Element not found.");
        }
        scanner.close();
    }
}
```

AD_1\Lab_7\FirstRepeatedElement.java

```
package AD_1.Lab_7;

import java.util.HashSet;
import java.util.Scanner;

public class FirstRepeatedElement {
    public static int findFirstRepeated(int[] arr) {
        HashSet<Integer> set = new HashSet<>();
        for (int num : arr) {
            if (!set.add(num)) {
                return num;
            }
        }
        return -1; // No repeated elements
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // System.out.print("Enter the number of elements: ");
```

```
        // int n = scanner.nextInt();
        int[] arr = {34,56,77,1,5,6,6,6,7,8,10,34,20,30};

        // System.out.println("Enter the elements:");
        // for (int i = 0; i < arr.length; i++) {
        //     arr[i] = scanner.nextInt();
        // }

        int result = findFirstRepeated(arr);
        if (result != -1) {
            System.out.println("First repeated element: " + result);
        } else {
            System.out.println("No repeated elements found.");
        }
        scanner.close();
    }
}
```

AD_1\Lab_7\MaxFrequencyElement.java

```
package AD_1.Lab_7;

public class MaxFrequencyElement  {
    public static int getMax(int arr[], int size) {
        int max = arr[0];
        int maxCount = 1;

        for (int i = 0; i < size; i++) {
            int count = 1;
            for (int j = i + 1; j < size; j++) {
                if (arr[i] == arr[j])
                    count++;
            }
            if (count > maxCount) {
                max = arr[i];
                maxCount = count;
            }
        }
        return max;
    }

    public static void main(String[] args) {
        int arr[] = {1, 6, 4, 2, 9, 2, 9, 5, 9, 1};
        int max = getMax(arr, arr.length);
        System.out.println("The element that appears maximum number of times: " + max);
    }
}
```

AD_1\Lab_7\MinMaxDifference.java

```
package AD_1.Lab_7;

import java.util.Arrays;
```

```java
import java.util.Scanner;

public class MinMaxDifference {
    public static void findMinMaxDifference(int[] arr) {
        Arrays.sort(arr);

        int minDiff = Integer.MAX_VALUE;
        int maxDiff = Integer.MIN_VALUE;

        for (int i = 1; i < arr.length; i++) {
            int diff = arr[i] - arr[i - 1];
            minDiff = Math.min(minDiff, diff);
            maxDiff = Math.max(maxDiff, diff);
        }

        System.out.println("Minimum difference: " + minDiff);
        System.out.println("Maximum difference: " + maxDiff);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        findMinMaxDifference(arr);
        scanner.close();
    }
}
```

AD_1\Lab_7\MissingNumber.java

```java
package AD_1.Lab_7;

public class MissingNumber {
    public static int findMissing(int arr[], int size) {
        for (int i = 0; i <= size; i++) { // Check numbers from 0 to size
            boolean found = false;
            for (int j = 0; j < size; j++) {
                if (arr[j] == i) {
                    found = true;
                    break;
                }
            }
            if (!found) {
                return i; // Return the first missing number
            }
        }
        return -1; // If no number is missing
    }
```

```
    public static void main(String[] args) {
        int arr[] = { 6, 8, 40, 2, 3, 5, 12, 0, 1 };
        System.out.println("The missing number is: " + findMissing(arr, arr.length));
    }
}
```

AD_1\Lab_7\PrintDuplicates.java

```
package AD_1.Lab_7;

import java.util.HashSet;
import java.util.Scanner;

public class PrintDuplicates {
    public static void printDuplicates(int[] arr) {
        HashSet<Integer> set = new HashSet<>();
        boolean foundDuplicate = false;

        for (int num : arr) {
            if (!set.add(num)) {
                System.out.println("Duplicate found: " + num);
                foundDuplicate = true;
            }
        }

        if (!foundDuplicate) {
            System.out.println("No duplicates found.");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        printDuplicates(arr);
        scanner.close();
    }
}
```

AD_1\Lab_8\LinkedListOperations.java

```
package AD_1.Lab_8;

public class LinkedListOperations {
    static class Node {
```

```
    int data;
    Node next;

    public Node(int data) {
        this.data = data;
        this.next = null;
    }
}

static class LinkedList {
    Node head;

    // Insert at the start
    public void insertAtStart(int data) {
        Node newNode = new Node(data);
        newNode.next = head;
        head = newNode;
    }

    // Insert at the end
    public void insertAtEnd(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
            return;
        }
        Node temp = head;
        while (temp.next != null) {
            temp = temp.next;
        }
        temp.next = newNode;
    }

    // Insert at a specific position
    public void insertAtPosition(int data, int position) {
        Node newNode = new Node(data);
        if (position == 1) {
            newNode.next = head;
            head = newNode;
            return;
        }
        Node temp = head;
        for (int i = 1; i < position - 1 && temp != null; i++) {
            temp = temp.next;
        }
        if (temp == null) {
            System.out.println("Position out of range");
            return;
        }
        newNode.next = temp.next;
        temp.next = newNode;
    }

    // Delete at the start
    public void deleteAtStart() {
        if (head != null) {
```

```
            head = head.next;
        }
    }

    // Delete at the end
    public void deleteAtEnd() {
        if (head == null) return;
        if (head.next == null) {
            head = null;
            return;
        }
        Node temp = head;
        while (temp.next != null && temp.next.next != null) {
            temp = temp.next;
        }
        temp.next = null;
    }

    // Delete at a specific position
    public void deleteAtPosition(int position) {
        if (head == null) return;
        if (position == 1) {
            head = head.next;
            return;
        }
        Node temp = head;
        for (int i = 1; temp != null && i < position - 1; i++) {
            temp = temp.next;
        }
        if (temp == null || temp.next == null) return;
        temp.next = temp.next.next;
    }

    // Print the list
    public void printList() {
        Node temp = head;
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.next;
        }
        System.out.println();
    }

    // Reverse the list
    public void reverse() {
        Node prev = null, current = head, next = null;
        while (current != null) {
            next = current.next;
            current.next = prev;
            prev = current;
            current = next;
        }
        head = prev;
    }
}
```

```
    public static void main(String[] args) {
        LinkedList list = new LinkedList();

        // Performing various operations
        list.insertAtEnd(1);
        list.insertAtEnd(2);
        list.insertAtEnd(3);
        list.insertAtStart(0);
        list.insertAtPosition(5, 3);

        System.out.print("Linked List after insertions: ");
        list.printList(); // Output: 0 1 5 2 3

        list.deleteAtStart();
        list.deleteAtEnd();
        list.deleteAtPosition(2);

        System.out.print("Linked List after deletions: ");
        list.printList(); // Output: 2

        list.reverse();
        System.out.print("Linked List after reversal: ");
        list.printList(); // Output: 2
    }
}
```

AD_1\Lab_9\Javacollection.java

```
package AD_1.Lab_9;

import java.util.ArrayList;
public class Javacollection {

    public static void main(String[] args) {
        ArrayList <Integer> list= new ArrayList <Integer> ();
        list.add(10);
        list.add(89);
        list.add(24);
        System.out.println(list);
        int element =list.get(0);
        System.out.println(element);
        list.add(1,34);
    }

}
```