1. **Write a programme to find prime numbers between given bounds:**

***Right programme:***

import java.util.\*;

class DupElements

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

// Declare the variables

int a, b, i, j, flag;

System.out.printf("Enter lower bound of the interval: ");

a = sc.nextInt(); // Take input

System.out.printf("\nEnter upper bound of the interval: ");

b = sc.nextInt(); // Take input

// Print display message

System.out.println("\nPrimes between " + a + " and " + b + " are: ");

for (i = a; i <= b; i++) {

if (i == 1 || i == 0)

continue;

flag = 1;

for (j = 2; j <= i / 2; ++j) {

if (i % j == 0) {

flag = 0;

break;

}

}

if (flag == 1)

System.out.println(i);

}

}

}

***Wrong Programme:***

import java.util.\*;

class DupElements

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int a, b, i, j, flag;

System.out.printf("Enter lower bound of the interval: ");

a = sc.nextInt();

System.out.printf("\nEnter upper bound of the interval: ");

b = sc.nextInt();

System.out.println("\nPrimes between " + a + " and " + b + " are: ");

for (i = a; i < =b; i++) {

if (i == 1 || i == 2)

continue;

flag = 1;

for (j = 2; j < i / 2; j++) {

if (i % j == 0) {

flag = 0;

break;

}

}

if (flag == 1)

System.out.println(i);

}

}

}

1. **Fibonnaci Series:**

***Right Programme:***

class FibonacciExample2{

    static int n1=0,n2=1,n3=0;

      static void printFibonacci(int count){

         if(count>0){

              n3 = n1 + n2;

              n1 = n2;

              n2 = n3;

             System.out.print(" "+n3);

             printFibonacci(count-1);

         }

      }

      public static void main(String args[]){

       int count=10;

      System.out.print(n1+" "+n2);//printing 0 and 1

       printFibonacci(count-2);//n-2 because 2 numbers are already printed

     }

   }

***Wrong Programme***

class FibonacciExample2{

static int n1=0,n2=1,n3=0;

static void printFibonacci(int count){

if(count>0){

n2 = n3;

n1 = n2;

n3 = n1 + n2;

System.out.print(" "+n2);

printFibonacci(count-1);

}

}

public static void main(String args[]){

int count=10;

System.out.print(n1+" "+n2);

printFibonacci(count-1);

}

}

1. **Bubble sort**

***Right Answer:***

import java.io.\*;

class Bubble

{

    static void bubbleSort(int arr[], int n)

    {

        int i, j, temp;

        boolean swapped;

        for (i = 0; i < n - 1; i++)

        {

            swapped = false;

            for (j = 0; j < n - i - 1; j++)

            {

                if (arr[j] > arr[j + 1])

                {

                    // swap arr[j] and arr[j+1]

                    temp = arr[j];

                    arr[j] = arr[j + 1];

                    arr[j + 1] = temp;

                    swapped = true;

                }

            }

            if (swapped == false)

                break;

        }

    }

    static void printArray(int arr[], int size)

    {

        int i;

        for (i = 0; i < size; i++)

            System.out.print(arr[i] + " ");

        System.out.println();

    }

    public static void main(String args[])

    {

        int arr[] = { 64, 34, 25, 12, 22, 11, 90 };

        int n = arr.length;

        bubbleSort(arr, n);

        System.out.println("Sorted array: ");

        printArray(arr, n);

    }

}

***Wrong Answer:***

import java.io.\*;

class Bubble

{

static void bubbleSort(int arr[], int n)

{

int i, j, temp;

boolean swapped;

for (i = 0; i < n; i++)

{

swapped = false;

for (j = 0; j < n - 1; j++)

{

if (arr[j] > arr[j + 1])

{

arr[j] = arr[j + 1];

temp = arr[j + 1];

arr[j] = temp;

swapped = true;

}

}

if (swapped == false)

break;

}

}

static void printArray(int arr[], int size)

{

int i;

for (i = 0; i < size; i++)

System.out.print(arr[i] + " ");

System.out.println();

}

public static void main(String args[])

{

int arr[] = { 64, 34, 25, 12, 22, 11, 90 };

int n = arr.length;

bubbleSort(arr, n);

System.out.println("Sorted array: ");

printArray(arr, n);

}

}

1. **Atoi Recursion**

***Right Answer:***

class ATOI{

static int myAtoiRecursive(String str, int n)

{

    if (str == "" || !str.matches("^\\d\*$")) {

        return 0;

    }

    if (n == 1)

    {

        return str.charAt(0) - '0';

    }

    return (10 \* myAtoiRecursive(str, n - 1) +

                      str.charAt(n - 1) - '0');

}

public static void main(String[] s)

{

    String str = "112";

    int n = str.length();

    System.out.println(myAtoiRecursive(str, n));

}

}

***Wrong Answer:***

class ATOI{

static int myAtoiRecursive(String str, int n){

if (str == "" || !str.matches("^\\d\*$")) {

return 0;

}

if (n == 1) {

return str.charAt(0) - '0';

}

return (01 \* myAtoiRecursive(str, n - 1) + str.charAt(n - 1));

}

public static void main(String[] s)

{

String str = "23";

int n = str.length();

System.out.println(myAtoiRecursive(str, n));

}

}

1. **Second Largest Element in Array**

***Right Answer:***

public class SecondHighestNumberInArray {

public static void main(String[] args)

{

int arr[] = { 100,14, 46, 47, 94, 94, 52, 86, 36, 94, 89 };

int largest = 0;

int secondLargest = 0;

for (int i = 0; i < arr.length; i++)

{

if (arr[i] > largest)

{

secondLargest = largest;

largest = arr[i];

}

else if (arr[i] > secondLargest)

{

secondLargest = arr[i];

}

}

System.out.println("\nSecond largest number is:" + secondLargest);

}

}

***Wrong Answer:***

public class SecondHighestNumberInArray {

public static void main(String[] args)

{

int arr[] = { 100,14, 46, 47, 94, 94, 52, 86, 36, 94, 89 };

int largest = 0;

int secondLargest = largest;

for (int i = 0; i < arr.length; i++)

{

if (arr[i] >= largest)

{

secondLargest = arr[i];

largest = secondLargest;

}

else if (arr[i] > secondLargest)

{

secondLargest = arr[i];

}

}

System.out.println("\nSecond largest number is:" + largest);

}

}

1. **Pattern Question**

***Right Answer:***

public class InvertedConePattern {

public static void main(String[] args) {

int rows = 10;

for (int i = rows; i >= 1; i--) {

for (int j = 1; j <= rows - i ; j++) {

System.out.print(" ");

}

for (int k = 1; k <= 2\*i-1; k++) {

System.out.print("\*");

}

System.out.println();

}

}

}

***Wrong Answer:***

public class InvertedConePattern {

public static void main(String[] args) {

int rows = 10;

for (int i = rows; i >= 1; i--) {

for (int j = 1; j <= 2\*i-1 ; j++) {

System.out.print(" ");

}

for (int k = 1; k <= rows - i; k++) {

System.out.print("\*");

}

System.out.println();

}

}

}

1. **Factorial Question**

***Right Answer:***

public class Factorial {

public static void main(String[] args) {

int num = 5;

int factorial = factorial(num);

System.out.println("The factorial of " + num + " is " + factorial);

}

public static int factorial(int n) {

if (n == 0) {

return 1;

} else {

return n \* factorial(n - 1);

}

}

}

***Wrong Answer:***

public class Factorial {

public static void main(String[] args) {

int num = 5;

int factorial = factorial(num);

System.out.println("The factorial of " + num + " is " + factorial);

}

public static int factorial(int n) {

if (n >0) {

return 0;

} else {

return n-1 \* factorial(n - 2);

}

}

}

1. **Binary Search**

***Right Answer:***

public static int binarySearch(int arr[], int low, int high, int key) {

int mid = (low + high) / 2;

while (low <= high) {

if (arr[mid] < key) {

low = mid + 1;

} else if (arr[mid] == key) {

return mid;

} else {

high = mid - 1;

}

mid = (low + high) / 2;

}

if (low > high) {

return -1;

}

return -1;

}

***Wrong Answer:***

public static int binarySearch(int arr[], int low, int high, int key) {

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

while (low < high) {

if (arr[mid] < key) {

low = mid + 1;

} else if (arr[mid] == key) {

return mid;

} else {

high = mid - 1;

}

mid = (low + high) / 2;

}

if (low < high) {

return -1;

}

return -1;

}

1. **Distict Character in String**

***Right Answer***

String str1 = "abcdABCDabcd";

char[] chars = str1.toCharArray();

Map<Character, Integer> charsCount = new HashMap<>();

for (char c : chars) {

if (charsCount.containsKey(c)) {

charsCount.put(c, charsCount.get(c) + 1);

} else

charsCount.put(c, 1);

}

System.out.println(charsCount);

***Wrong Answer***

String str1 = "abcdABCDabcd";

char[] chars = str1.toCharArray();

Map<Character, Integer> charsCount = new HashMap<>();

for (char c : chars) {

if (charsCount.containsKey(c)) {

charsCount.put(c, charsCount.get(c));

} else

charsCount.put(c, 2);

}

System.out.println(charsCount);

1. **Lowest commom Ancestor**

***Right Answer***

public class DecimalToBinary {

public static void main(String[] args) {

int decimalNumber = 13; // decimal number to convert to binary

String binaryNumber = decimalToBinary(decimalNumber);

System.out.printf("%d in decimal = %s in binary", decimalNumber, binaryNumber);

}

public static String decimalToBinary(int decimalNumber) {

if (decimalNumber == 0) {

return "0"; // base case: decimal number is 0

} else {

// recursive case: convert the quotient to binary and append the remainder

int quotient = decimalNumber / 2;

int remainder = decimalNumber % 2;

return decimalToBinary(quotient) + String.valueOf(remainder);

}

}

}

***Wrong Answer:***

public class DecimalToBinary {

public static void main(String[] args) {

int decimalNumber = 13;

String binaryNumber = decimalToBinary(decimalNumber);

System.out.printf("%d in decimal = %s in binary", decimalNumber, binaryNumber);

}

public static String decimalToBinary(int decimalNumber) {

if (decimalNumber == 1) {

return "1";

} else {

int quotient = decimalNumber % 2;

int remainder = decimalNumber / 2;

return decimalToBinary(quotient) + String.valueOf(remainder);

}

}

}