Sample java Programs

Name : R Anush

Q1: How do you check if a list of integers contains only odd numbers in Java?

Input:

```
import java.util.List;
public class OddNumberChecker {
        public static boolean containsOnlyOddNumbers(List<Integer> numbers)
{
      for (int num : numbers) {
         if (num % 2 == 0) {
            return false; // If even number found, return false
         }
      }
      return true; // Only odd numbers found
   }

   public static void main(String[] args) {
      // Example list
      List<Integer> myList = List.of(3, 5, 7, 9);

      // Check if the list contains only odd numbers
      boolean result = containsOnlyOddNumbers(myList);
      System.out.println("List contains only odd numbers: " + result);
   }

}
```

Output:

```
List contains only odd numbers: true
```

Q2: How do you check whether a string (can be number also) is a palindrome in Java?

Input:

```
public class PalindromeChecker {

        public static boolean isPalindrome(String str)
        {
              String reversedStr = "";
              for (int i = str.length() - 1; i >= 0; i--) {
                 reversedStr += str.charAt(i);
              }
              return str.equals(reversedStr);
           }

           public static void main(String[] args) {
              String str1 = "madam";
              String str2 = "racecar";
              String str3 = "hello";

              System.out.println("Is \"" + str1 + "\" a palindrome? " +
isPalindrome(str1));
              System.out.println("Is \"" + str2 + "\" a palindrome? " +
isPalindrome(str2));
              System.out.println("Is \"" + str3 + "\" a palindrome? " +
isPalindrome(str3));
           }
        }
```

Output:

```
Is "madam" a palindrome? true
Is "racecar" a palindrome? true
Is "hello" a palindrome? false
```

Q3 : How do you remove leading and trailing spaces from a string in Java?

<u>Input:</u>

```
public class RemoveSpaces {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
String stringWithSpaces = "   Hello, there!   ";

      // Remove leading and trailing spaces using trim()
      String trimmedString = stringWithSpaces.trim();

      System.out.println("Original string: '" + stringWithSpaces + "'");
      System.out.println("String after removing leading and trailing spaces: '" +
trimmedString + "'");
         }

}
```

<u>Output:</u>

```
Original string: '   Hello, there!   '
String after removing leading and trailing spaces: 'Hello, there!'
```

Q4: How can you find the factorial of an integer in Java?

Input:

```
public class Factorial {
        public static long factorial(int n) {
      if (n < 0) {
         throw new IllegalArgumentException("Factorial is not defined for negative numbers");
      }
      long result = 1;
      for (int i = 1; i <= n; i++) {
         result *= i;
      }
      return result;
   }

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                 int number = 5; // Change this number to calculate factorial for a different value
              long fact = factorial(number);
              System.out.println("Factorial of " + number + " is: " + fact);
         }

}
```

Output:

```
Factorial of 5 is: 120
```

Q5 : Write a Java program that illustrates merge sort.

Input:

```java
package hackerRank;

public class MergeSort {

      public static void merge(int[] arr, int left, int middle, int right) {
    // Calculate sizes of two subarrays to be merged
    int n1 = middle - left + 1;
    int n2 = right - middle;

    // Create temporary arrays
    int[] leftArray = new int[n1];
    int[] rightArray = new int[n2];

    // Copy data to temporary arrays
    for (int i = 0; i < n1; ++i) {
        leftArray[i] = arr[left + i];
    }
    for (int j = 0; j < n2; ++j) {
        rightArray[j] = arr[middle + 1 + j];
    }

    // Merge the temporary arrays

    // Initial indexes of first and second subarrays
    int i = 0, j = 0;

    // Initial index of merged subarray array
    int k = left;
    while (i < n1 && j < n2) {
        if (leftArray[i] <= rightArray[j]) {
            arr[k] = leftArray[i];
            i++;
        } else {
            arr[k] = rightArray[j];
            j++;
        }
        k++;
    }
```

```
        // Copy remaining elements of leftArray[], if any
        while (i < n1) {
            arr[k] = leftArray[i];
            i++;
            k++;
        }

        // Copy remaining elements of rightArray[], if any
        while (j < n2) {
            arr[k] = rightArray[j];
            j++;
            k++;
        }
    }

    public static void mergeSort(int[] arr, int left, int right) {
        if (left < right) {
            // Find the middle point
            int middle = (left + right) / 2;

            // Sort first and second halves
            mergeSort(arr, left, middle);
            mergeSort(arr, middle + 1, right);

            // Merge the sorted halves
            merge(arr, left, middle, right);
        }
    }

    public static void printArray(int[] arr) {
        for (int value : arr) {
            System.out.print(value + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int[] arr = {12, 11, 13, 5, 6, 7};
        System.out.println("Original array:");
        printArray(arr);

        mergeSort(arr, 0, arr.length - 1);
        System.out.println("Sorted array:");
        printArray(arr);
    }

}
```

Output:

```
Original array:
12 11 13 5 6 7
Sorted array:
5 6 7 11 12 13
```

Q6: Write Java program that checks if two arrays contain the same elements.

Input:

```java
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;

public class ArrayEqualityChecker {

        public static boolean arraysContainSameElements(int[] arr1, int[] arr2)
{
        if (arr1.length != arr2.length) {
            return false; // If lengths are different, arrays can't contain the same
elements
        }
        Set<Integer> set1 = new HashSet<>();
        Set<Integer> set2 = new HashSet<>();
        for (int value : arr1) {
            set1.add(value);
        }
        for (int value : arr2) {
            set2.add(value);
        }
        return set1.equals(set2); // Check if the sets are equal
    }
    public static void main(String[] args) {
        int[] array1 = {1, 2, 3, 4, 5};
        int[] array2 = {5, 4, 3, 2, 1};

        boolean result = arraysContainSameElements(array1, array2);
        System.out.println("Do the arrays contain the same elements? " + result);
    }
}
```

Output:

```
Do the arrays contain the same elements? True
```

Q7: How do you get the sum of all elements in an integer array in Java?

Input:

```
public class ArraySum {

      public static int getArraySum(int[] arr) {
        int sum = 0;
        for (int value : arr) {
           sum += value;
        }
        return sum;
    }
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5};

        int sum = getArraySum(array);
        System.out.println("Sum of the array elements: " + sum);
    }
}
```

Output:

```
Sum of the array elements: 15
```

Q8: Can you prove that a String object in Java is immutable programmatically?

Input:

```
public class StringImmutability {
      public static void main(String[] args) {
              // TODO Auto-generated method stub
              String str = "Hello";
        System.out.println("Original String: " + str);

        // Concatenating another string to the original one
        str = str.concat(" World");
        System.out.println("Modified String: " + str);
      }
}
```

Output:

```
Original String: Hello
Modified String: Hello World
```

Q9:Write a program to convert Decimal number to Binary number ?

Input:

```
public class DecimalToBinary {
        public static void decimalToBinary(int decimal) {
      int[] binaryArray = new int[32]; // Array to store binary digits (maximum
of 32 bits)

      int index = 0;
      while (decimal > 0) {
         binaryArray[index++] = decimal % 2; // Store the remainder (binary
digit) in the array
         decimal /= 2; // Divide the decimal number by 2 for next iteration
      }

      System.out.print("Binary representation: ");
      // Print the binary digits in reverse order (from the array)
      for (int i = index - 1; i >= 0; i--) {
         System.out.print(binaryArray[i]);
      }
   }

   public static void main(String[] args) {
      int decimalNumber = 25; // Change this number to convert a different
decimal number to binary

      System.out.println("Decimal number: " + decimalNumber);
      decimalToBinary(decimalNumber);
   }
}
```

Output:

```
Decimal number: 25
Binary representation: 11001
```

Q10: Write a program to do the matrix addition by reading the elements.

Input:

```
package hackerRank;
import java.util.Scanner;
public class MatrixAddition {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of rows for the matrices: ");
        int rows = scanner.nextInt();

        System.out.print("Enter the number of columns for the matrices: ");
        int columns = scanner.nextInt();

        int[][] matrix1 = new int[rows][columns];
        int[][] matrix2 = new int[rows][columns];

        // Reading elements for the first matrix
        System.out.println("Enter elements for the first matrix:");
        readMatrixElements(scanner, matrix1);

        // Reading elements for the second matrix
        System.out.println("Enter elements for the second matrix:");
        readMatrixElements(scanner, matrix2);

        // Performing matrix addition
        int[][] sumMatrix = addMatrices(matrix1, matrix2);

        // Displaying the result of matrix addition
        System.out.println("Resultant matrix after addition:");
        displayMatrix(sumMatrix);
    }

    public static void readMatrixElements(Scanner scanner, int[][] matrix) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                System.out.print("Enter element at position [" + i + "][" + j + "]: ");
                matrix[i][j] = scanner.nextInt();
```

```java
            }
        }
    }

    public static int[][] addMatrices(int[][] matrix1, int[][] matrix2) {
        int rows = matrix1.length;
        int columns = matrix1[0].length;
        int[][] sumMatrix = new int[rows][columns];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        return sumMatrix;
    }

    public static void displayMatrix(int[][] matrix) {
        for (int[] row : matrix) {
            for (int value : row) {
                System.out.print(value + " ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
Enter the number of rows for the matrices: 3
Enter the number of columns for the matrices: 3
Enter elements for the first matrix:
Enter element at position [0][0]: 2 3
Enter element at position [0][1]: Enter element at position [0][2]: 1 3
Enter element at position [1][0]: Enter element at position [1][1]: 1 1
Enter element at position [1][2]: Enter element at position [2][0]: 1 0
Enter element at position [2][1]: Enter element at position [2][2]: 17 39
Enter elements for the second matrix:
Enter element at position [0][0]: Enter element at position [0][1]: 33 45
Enter element at position [0][2]: Enter element at position [1][0]: 23 44
Enter element at position [1][1]: Enter element at position [1][2]: 1 9
Enter element at position [2][0]: Enter element at position [2][1]: 56 5
Enter element at position [2][2]: Resultant matrix after addition:
41 36 46
26 45 2
10 56 22
```