Assignment No. 7.

PAGE NO.
DATE:

Que:1 Answer:-

```
import java.util.HashMap;
import java.util.Map;
public class IsomorphicStrings {
    public static boolean isIsomorphic
                        (String s, String t)
    { if (s.length() != t.length()) {
        return false;
    }
Map<Character, Character> charMap = new HashMap
                                    <>();
Map<Character, Boolean> mapped = new HashMap
                                 <>();
for (int i=0; i<s.length(); i++){
    {
        char c1 = s.charAt(i);
        char c2 = t.charAt(i);
        if (charMap.containKey(c1)) {
        if (charMap.get(c1) != c2) {
            }
        } else {
            if (mapped.containsKey(c2)) {
            return false;
            }
        charMap.put(c1, c2);
        mapped.put(c2, true);
    }
    }
    return true;
    }
    public static void main (String[] args) {
        String s = "egg";
        String t = "add";
        boolean result = isIsomorphic(s,t);
        System.out.println(result);
    }
}
```

Que 2 Answer

```
import java.util.HashMap;
public class StrobogrammaticNumber {
    public boolean isStrobogrammatic(String num) {
        HashMap<Character, Character> map
                = new HashMap<>();
        map.put('0', '0');
        map.put('1', '1');
        map.put('6', '9');
        map.put('8', '8');
        map.put('9', '6');

        int left = 0;
        int right = num.length() - 1;

        while (left <= right) {
            char leftChar = num.charAt(left);
            char rightChar = num.charAt(right);

            if (!map.containsKey(leftChar) || map.get
                    (leftChar) != rightChar) {
                return false;
            }
            left;
            right--;
        }
        return true;
    }
    public static void main(String[] args) {
        StrobogrammaticNumber sn = new
                StrobogrammaticNumber();
        String num = "69";
        System.out.println(sn.isStrobogrammatic(num));
    }
}
```

Que 3 Answer:-

```
public class AddStrings {
    public String addStrings(String num1, String num2) {
        StringBuilder result = new StringBuilder();
        int carry = 0;
        int i = num1.length() - 1;
        int j = num2.length() - 1;

        while (i >= 0 || j >= 0 || carry > 0) {
            int digit1 = i >= 0 ? num1.charAt(i) - '0' : 0;
            int digit2 = j >= 0 ? num2.charAt(j) - '0' : 0;

            int sum = digit1 + digit2 + carry;
            carry = sum / 10;
            int digit = sum % 10;
            result.insert(0, digit);

            i--;
            j--;
        }
        return result.toString();
    }
    public static void main(String[] args) {
        AddStrings as = new AddStrings();

        String num1 = "11";
        String num2 = "123";

        System.out.println(as.addStrings(num1, num2));
    }
}
```

Que 4:- Answer:-

```
public class ReverseWordsInString {
public String reverseWords(String s) {
String[] words = s.split(" ");
StringBuilder result = new StringBuilder();
for(String word : words) {
StringBuilder reversedWord = new StringBuilder
(word);
result.append(reversedWord.reverse
()).append(" ");
}
return result.toString().trim();
}public static void main(String[] aegr) {
ReverseWordsInString swir = new ReverWords
InString();
String s = "Let's take LeetCode contest";
System.out.println(swir.reverseWords(s));
}
}
```

Ques:- Answer :-

```
public class ReverseString II {
public string reverseStr(String s, int k) {

char[] chars = s.toCharArray();
int n = chars.length;
int i = 0;

while (i < n) {
    int j = Math.min(i+k-1, n-1);
   reverse(chars, i, j);
    i += 2 * k;
}
return String.valueOf(chars);
{
private void reverse(char[] chars, int start,
                     int end) {
while (start < end) {
  char temp = chars[start];
  chars[start] = chars[end];
  chars[end] = temp;
  start++;
  end--;
{
}
public static void main(String[] args) {
 ReverseStringII rs = new ReverseStringII();
String s = "abcdefg";
 int k = 2;
System.out.println(rs.reverseStr(s,k));
                    Osgood;
{
}
```

```
public class RotateString {
public boolean rotateString(String s, String goal) {
if (s.length() != goal.length()) {
    return false;
}
    String doubleS = s+s;
    return doubleS.contains(goal);
}
public static void main(String[] args) {
RotateString rs = new RotateString();
String s = "abcde";
String goal = "cdeab";
System.out.println(rs.rotateString(s, goal));
}
}
```

Que 7 :- Answer :-

```
public class BackspaceStringCompare {
public boolean backspaceCompare
                    (String s, String t) {
return processingString(s).equals(processing
                    String(t));
}
private String processString (String str) {
 StringBuilder result = new StringBuilder();
for(char ch: str.toCharArray()) {
    if(ch != '#') {
        result.append(ch);
  } else if (result.length() > 0) {
    result.deleteCharAt(result.length()-1);
    }
}
return result.toString();
}
public static void main(String [] args) {
BackspaceStringCompare bsc = new
                    BackspaceStringCompar
String s = "ab#c";                  ();
String t = "ad#c";
String System.out.println(bsc.backspace
                    Compare(s,t));
{
}
```

Que 8 : Answer :-

```
public class CheckStraightLine {
public boolean checkStraightLine(int[][] coordinates)
{ int x0 = coordinates[0][0];
int y0 = coordinates[0][1];
int x1 = coordinates[1][0];
int y1 = coordinates[1][1];

for(int i=2; i<coordinates.length; i++) {
    int xi = coordinates[i][0];
    int yi = coordinates[i][1];

if((y1-y0)*(xi-x0) != (yi-y0)*(x1-x0)) {
        return false;
    }
}
return true;
}
public static void main(String[] args) {
CheckStraightLine csl = new CheckStraightLine();

int[][] coordinates = {{1,2},{2,3},{3,4},{4,5}
                       {5,6},{6,7}};

System.out.println(csl.checkStraightLine
                   (coordinates));

}
}
```