Q1.

public class Matrix3D {

    public static void set(int value, int indexX, int indexY, int indexZ, int[] array, int X, int Y, int Z) {

        int index = indexX \* (Y \* Z) + indexY \* Z + indexZ;

        array[index] = value;

    }

    public static int get(int indexX, int indexY, int indexZ, int[] array, int X, int Y, int Z) {

        int index = indexX \* (Y \* Z) + indexY \* Z + indexZ;

        return array[index];

    }

    public static void main(String[] args) {

        int X = 3, Y = 3, Z = 3;

        int[] array = new int[X \* Y \* Z];

        set(10, 0, 0, 0, array, X, Y, Z);

        set(20, 1, 1, 1, array, X, Y, Z);

        set(30, 2, 2, 2, array, X, Y, Z);

        System.out.println("Value at (0, 0, 0): " + get(0, 0, 0, array, X, Y, Z));

        System.out.println("Value at (1, 1, 1): " + get(1, 1, 1, array, X, Y, Z));

        System.out.println("Value at (2, 2, 2): " + get(2, 2, 2, array, X, Y, Z));

    }

}

Q2.

import java.util.Scanner;

public class Pwd {

    public static boolean isValid(String p){

        boolean valid = true;

        if(p.length()<=5 || p.length()>=12){

            valid = false;

            System.out.println("It's length is not between 5 & 12!");

            return valid;

        }

        if(!p.toLowerCase().equals(p)){

            valid = false;

            System.out.println("It has an uppercase letter!");

            return valid;

        }

        boolean hasDigit = false;

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

            if((int)p.charAt(i)>=48 && (int)p.charAt(i)<=57){

                hasDigit = true;

            }

            if(((int)p.charAt(i)>=33 && (int)p.charAt(i)<=47) || ((int)p.charAt(i)>=58 && (int)p.charAt(i)<=64)){

                valid = false;

                System.out.println("It has a special character!");

                return valid;

            }

        }

        if(hasDigit==false){

            valid = false;

            System.out.println("It does not have a digit!");

            return valid;

        }

        return valid;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the password: ");

        String p = sc.nextLine();

        System.out.println(isValid(p));;

    }

}

Q3.

import java.util.HashMap;

import java.util.Scanner;

public class CheckString {

    public static void exceedNum(String s, int n){

        StringBuilder sb = new StringBuilder();

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

            int count = 1;

            for(int j=i+1;j<s.length();j++){

                if(s.charAt(i) == s.charAt(j)){

                    count++;

                }

            }

            if(count>n){

                if(!sb.toString().contains(Character.toString(s.charAt(i)))){

                    sb.append(s.charAt(i));

                }

            }

        }

        System.out.println(sb.toString());

    }

    public static void numExceed(String s, int n){

        HashMap<Character,Integer> hm = new HashMap<>();

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

            if(hm.containsKey(s.charAt(i))){

                int val = hm.get(s.charAt(i));

                hm.put(s.charAt(i),val+1);

            } else{

                hm.put(s.charAt(i), 1);

            }

        }

        for(Character c:hm.keySet()){

            if(hm.get(c)>n){

                System.out.print(c);

            }

        }

        System.out.println();

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter string: ");

        String s = sc.nextLine();

        System.out.print("Enter n: ");

        int n = sc.nextInt();

        exceedNum(s, n);

    }

}

Q4.

import java.util.Scanner;

public class PerfectString {

    public static boolean checkPerfect(String s){

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

            int count = 1;

            for(int j = i+1;j<s.length();j++){

                if(s.charAt(i) == s.charAt(j)){

                    count++;

                }

            }

            if(count>1){

                return false;

            }

        }

        for(char i = 'a';i<='z';i++){

            if(!s.contains(Character.toString(i))){

                return false;

            }

        }

        return true;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a string: ");

        String s = sc.nextLine();

        System.out.println(checkPerfect(s));

    }

}

Q6.

interface Function {

    int evaluate(int x);

}

class Half implements Function {

    @Override

    public int evaluate(int x) {

        return x / 2;

    }

}

public class FUNCTION {

    public static int[] halveArray(int[] inputArray) {

        Function half = new Half();

        int[] resultArray = new int[inputArray.length];

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

            resultArray[i] = half.evaluate(inputArray[i]);

        }

        return resultArray;

    }

    public static void main(String[] args) {

        int[] inputArray = {10, 20, 30, 40, 50};

        int[] halvedArray = halveArray(inputArray);

        System.out.print("Halved values: ");

        for (int val : halvedArray) {

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

        }

    }

}