1. Bob has a string **S** of size **N**. He wants to create a transformed string by replicating the first character of **S** across the entire length of **S**.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

 public class Solution {

    public static String Replicate(int N, String S) {

        // Write your code here

        char ch=S.charAt(0);

        String out="";

        for(int i=0;i<N;i++)

        {

            out=out+ch;

        }

        return out;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int N = Integer.parseInt(scan.nextLine().trim());

        String S = scan.nextLine();

        String result = Replicate(N, S);

        System.out.println(result);

    }

}

2) Joe has two integers **a** and **b**. He wants to find the **remainder** when **a** is divided by **b** and do the following:

1. If remainder is greater than 1 then output the sum of a, b and remainder.
2. If remainder is 0 or 1 then output the sum of a and remainder.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

 public class Solution {

    public static int remSum(int a, int b) {

    int sum=0;

    if((a%b)>1)

    {

            sum=a+b+(a%b);

    }

    else

    {

      sum=a+(a%b);

    }

    return sum;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int a = Integer.parseInt(scan.nextLine().trim());

        int b = Integer.parseInt(scan.nextLine().trim());

        int result = remSum(a, b);

        System.out.println(result);

    }

}

1. Bob has a string **S**. He wants to replace all vowels in **S** with the number 3.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static String ReplaceVowels(String S) {

        // Write your code here

        String out="";

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

        {

            char ch = S.charAt(i);

           if (ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U'||ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u')

                {

                    out=out+'3';

                }

            else

            {

                out=out+ch;

            }

        }

        return(out);

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String S = scan.nextLine();

        String result = ReplaceVowels(S);

        System.out.println(result);

    }

}

4) Jake has a string array **arr** of size **n** and a special character **target.** He wants to count the total number of occurrences of **target** in all strings in **arr.**

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static int buggyCountTargetCharacter(int n, List<String> arr, char target) {

        // Fix the code here

        int count = 0;

        for (int i = 0; i < n; i++) {

            String str = arr.get(i);

            for (int j = 0; j < str.length(); j++) {

                if (str.charAt(j) == target) {

                    count++;

                }

            }

        }

        return count;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = Integer.parseInt(scan.nextLine().trim());

        List<String> arr = new ArrayList<>(n);

        for(int j=0; j<n; j++) {

            arr.add(scan.nextLine());

        }

        char target = scan.nextLine().charAt(0);

        int result = buggyCountTargetCharacter(n, arr, target);

        System.out.println(result);

    }

}

1. John has an array **arr** of size **n**. He wants to find the sum of squares of all even numbersin **arr**.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static int buggySumOfEvenNumbers(int n, List<Integer> arr) {

        // Fix the code here

        int sum = 0;

        for (int i = 0; i < n; i++) {

            if (arr.get(i) % 2 == 0) {

                sum += arr.get(i);

            }

        }

         return sum;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = Integer.parseInt(scan.nextLine().trim());

        List<Integer> arr = new ArrayList<>(n);

        for(int j=0; j<n; j++) {

            arr.add(Integer.parseInt(scan.nextLine().trim()));

        }

        int result = buggySumOfEvenNumbers(n, arr);

        System.out.println(result);

    }

}

Level 2

6)You are given a partially implemented code to check if a given string is a palindrome. If the given string is a palindrome return 1 else return 0.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static boolean buggyIsPalindrome(String s) {

        // Fix the code here

        String cleanedS = s.replaceAll("[^A-Za-z0-9]+", "").toLowerCase();

        for(int i = 0; i < cleanedS.length() / 2; i++) {

            if(cleanedS.charAt(i) != cleanedS.charAt(cleanedS.length() - i - 1)) {

                return false;

            }

        }

        return true;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String s = scan.nextLine();

        boolean result = buggyIsPalindrome(s);

        System.out.println(result? "1": "0");

    }

}

7) John has an array **arr** of size **n**. He wants to find the sum of squares of all odd numbersin **arr**.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static boolean isOdd(int num) {

        return num % 2 != 0;

    }

    public static int buggySumOfOddSquares(int n, List<Integer> arr) {

        // Fix the code here

        int sum = 0;

        for (int i = 0; i < n; i++) {

            if (isOdd(arr.get(i))) {

                sum += arr.get(i) \* arr.get(i);

            }

        }

        return sum;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = Integer.parseInt(scan.nextLine().trim());

        List<Integer> arr = new ArrayList<>(n);

        for(int j=0; j<n; j++) {

            arr.add(Integer.parseInt(scan.nextLine().trim()));

        }

        int result = buggySumOfOddSquares(n, arr);

        System.out.println(result);

    }

}

8) You are given a partially implemented code to determine if a given string is a valid email address. The code reads a string and calls the function buggyIsValidEmail to check if the string is a valid email address.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Solution {

    public static boolean buggyIsValidEmail(String email) {

        // Fix the code here

        Pattern pattern = Pattern.compile("^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{3,}$");

        Matcher matcher = pattern.matcher(email);

        return matcher.matches();

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String email = scan.nextLine();

        boolean result = buggyIsValidEmail(email);

        System.out.println(result? "true" : "false");

    }

}

9) You are given a partially implemented code to determine if a given string is a valid URL. The code reads a string and calls the function buggyIsValidURL to check if the string is a valid URL.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

import java.util.regex.Pattern;

import java.util.regex.Matcher;

public class Solution {

    public static boolean buggyIsValidURL(String url) {

        // Fix the code here

        Pattern pattern = Pattern.compile("^(((https?|http|ftp|smtp))+://)+[a-z0-9]+([.a-z0-9])+[.a-z]{2,}(/\\S\*)?$", Pattern.CASE\_INSENSITIVE);

        Matcher matcher = pattern.matcher(url);

        return matcher.matches();

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String url = scan.nextLine();

        boolean result = buggyIsValidURL(url);

        System.out.println(result? "true": "false");

    }

}

10) Jill has a string **str**. She wants to find the number of words in **str** that start with a specified character **c**.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static int buggyCountWordsStartingWithCharacter(char c, String str) {

        // Fix the code here

        int count = 0;

        String[] words = str.split(" ");

        for (String word : words) {

            if (word.length() > 0 & word.charAt(0) == c) {

                count++;

            }

        }

        return count;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        char c = scan.nextLine().charAt(0);

        String str = scan.nextLine();

        int result = buggyCountWordsStartingWithCharacter(c, str);

        System.out.println(result);

    }

}

11) You are given a partially implemented code to determine if a given string is a valid email address. The code reads a string and calls the function buggyIsValidEmail to check if the string is a valid email address.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Solution {

    public static boolean buggyIsValidEmail(String email) {

        // Fix the code here

        Pattern pattern = Pattern.compile("^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{3,}$");

        Matcher matcher = pattern.matcher(email);

        return matcher.matches();

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String email = scan.nextLine();

        boolean result = buggyIsValidEmail(email);

        System.out.println(result? "true" : "false");

    }

}

12) You are given a partially implemented code to determine if a given string is a valid URL. The code reads a string and calls the function buggyIsValidURL to check if the string is a valid URL.

However, the function contains a bug that causes it to produce incorrect results.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

import java.util.regex.Pattern;

import java.util.regex.Matcher;

public class Solution {

    public static boolean buggyIsValidURL(String url) {

        // Fix the code here

        Pattern pattern = Pattern.compile("^((https?|ftp|smtp)://)?[a-z0-9]+([-.][a-z0-9]+)\*\\.[a-z]{2,}(/\\S\*)?$", Pattern.CASE\_INSENSITIVE);

        Matcher matcher = pattern.matcher(url);

        return matcher.matches();

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String url = scan.nextLine();

        boolean result = buggyIsValidURL(url);

        System.out.println(result? "true": "false");

    }

}

13) You are given a partially implemented code to determine if a given string contains all unique characters. The code reads a string and calls the function buggyHasUniqueCharacters to check if the string has all unique characters. However, the function contains a bug that causes it to produce incorrect results.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static boolean buggyHasUniqueCharacters(String str) {

        // Fix the code here.

        boolean[] charSet = new boolean[26];

        for (char c : str.toCharArray()) {

            if (charSet[c - 'a']) {

                return false;

            }

            charSet[c - 'a'] = true;

        }

        return true;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String str = scan.nextLine();

        boolean result = buggyHasUniqueCharacters(str);

        System.out.println(result? "1": "0");

    }

}

14) Jake has a string array **arr** of size **n** and an integer value **length.** He wants to count the total number strings in **arr** whose length is equal to the given integer value**.**

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static int buggyCountStringsOfTargetLength(int n, List<String> arr, int length) {

        // Fix the code here

        int count = 0;

        for (int i = 0; i < n; i++) {

            if (arr.get(i).length() == length) {

                count++;

            }

        }

        return count;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = Integer.parseInt(scan.nextLine().trim());

        List<String> arr = new ArrayList<>(n);

        for(int j=0; j<n; j++) {

            arr.add(scan.nextLine());

        }

        int length = Integer.parseInt(scan.nextLine().trim());

        int result = buggyCountStringsOfTargetLength(n, arr, length);

        System.out.println(result);

    }

}

15) John has an array **arr** of size **n**. He wants to find the **sum of squares** of all even numbers in **arr**.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static int buggySumOfEvenSquares(int n, List<Integer> arr) {

        // Fix the bug here

        int sum = 0;

        for (int i = 0; i < n; i++) {

            int num = arr.get(i);

            if (num % 2 == 0) {

                sum += num \* num;

            }

        }

        return sum;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = Integer.parseInt(scan.nextLine().trim());

        List<Integer> arr = new ArrayList<>(n);

        for(int j=0; j<n; j++) {

            arr.add(Integer.parseInt(scan.nextLine().trim()));

        }

        int result = buggySumOfEvenSquares(n, arr);

        System.out.println(result);

    }

}

16) Jack has two strings **str** and **t**. He wants to find the total number of occurrences of string **t** in **str.**

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static int buggyCountSubstringOccurrences(String str, String t) {

        // Fix the code here

        int count = 0;

        int index = 0;

        while ((index = str.indexOf(t, index)) >= 0) {

            count++;

            index += t.length();

        }

        return count;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String str = scan.nextLine();

        String t = scan.nextLine();

        int result = buggyCountSubstringOccurrences(str, t);

        System.out.println(result);

    }

}

17) Sam has an array **arr** containing **n** integers. He wants to calculate the product of all prime numbers in **arr**.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static int buggyProductOfPrimeNumbers(int n, List<Integer> arr) {

        // Fix the code here

        int product = 1;

        for (int i = 0; i < n; i++) {

            int num = arr.get(i);

            if (num < 2) {

                continue;

            }

            boolean isPrime = true;

            for (int j = 2; j \* j <= num; j++) {

                if (num % j == 0) {

                    isPrime = false;

                    break;

                }

            }

            if (isPrime) {

                product \*= num;

            }

        }

        return product;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = Integer.parseInt(scan.nextLine().trim());

        List<Integer> arr = new ArrayList<>(n);

        for(int j=0; j<n; j++) {

            arr.add(Integer.parseInt(scan.nextLine().trim()));

        }

        int result = buggyProductOfPrimeNumbers(n, arr);

        System.out.println(result);

    }

}

18) You are given a partially implemented code to determine if a given string is a valid credit card number. The code reads a string and calls the function buggyIsValidCreditCard to check if the string is a valid credit card number.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static boolean buggyIsValidCreditCard(String number) {

        // Fix the code here

        int len = number.length();

        if (len != 16) return false;

        int sum = 0;

        boolean doubleNext = true;

        for (int i = 0;i<len;i++) {

            char c = number.charAt(i);

            if (c < '0' || c > '9') return false;

            int digit = c - '0';

            if (doubleNext) {

                digit \*= 2;

                if (digit > 9) digit = digit/10 + digit%10;

            }

            sum += digit;

            doubleNext = !doubleNext;

        }

        return sum % 10 == 0;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String number = scan.nextLine();

        boolean result = buggyIsValidCreditCard(number);

        System.out.println(result? "true": "false");

    }

}

19) John has an array **arr** of size **n**. He wants to find the sum of the factorial of all odd numbers in **arr**.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

public class Solution {

    public static long buggySumOfOddFactorials(int n, List<Integer> arr) {

        // Fix the code here

        long sum = 0;

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

            if (arr.get(i) % 2 == 1) {

                long fact = 1;

                for (int j = 1; j <= arr.get(i); j++) {

                    fact \*= j;

                }

                sum += fact;

            }

        }

        return sum;

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = Integer.parseInt(scan.nextLine().trim());

        List<Integer> arr = new ArrayList<>(n);

        for(int j=0; j<n; j++) {

            arr.add(Integer.parseInt(scan.nextLine().trim()));

        }

        long result = buggySumOfOddFactorials(n, arr);

        System.out.println(result);

    }

}

20) You are given a partially implemented code to determine if a given string is a valid phone number. The code reads a string and calls the function buggyIsValidPhoneNumber to check if the string is a valid phone number.

import java.io.\*;

import java.util.\*;

import java.lang.Math;

import java.util.Scanner;

import java.util.regex.Pattern;

import java.util.regex.Matcher;

public class Solution {

    public static boolean buggyIsValidPhoneNumber(String phoneNumber) {

        // Fix the code here

        Pattern pattern = Pattern.compile("^([0-9]{3}-?[0-9]{3}-?[0-9]{4})$");

        Matcher matcher = pattern.matcher(phoneNumber);

        return matcher.matches();

    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        String phoneNumber = scan.nextLine();

        boolean result = buggyIsValidPhoneNumber(phoneNumber);

        System.out.println(result? "true": "false");

    }

}