COMPETITIVE PROGRAMMING

Real Competition C-PROM2016

Computer Science Dept, UiTM PERLIS

PROBLEM A: MULTIPLICATION TABLE

Produce a multiplication table based on user input.

Input
Integer number in range 1 to 12 only. Validate the input.

Output
The output will display the multiplication table.

Sample Input	Sample Output
3	1*3=3 2*3=6 3*3=9 4*3=12 5*3=15 6*3=18 7*3=21 8*3=24 9*3=27 10*3=30 11*3=33 12*3=36

Problem setter: UMI HANIM MAZLAN

PROBLEM B: SPLIT & COUNT

You are required to split a sentence into individual words and count the length for each word. However, if there are any semicolon (;) or a dash (-) in a particular sentence, it will not be counted as part of the word, but it will accept any other characters.

Input
User has to input a sentence from a standard input.

Output
The output must be formatted as given in the sample output below and will omit if there is any semicolon (;) or a dash (-) in the sentence. You need to count and display how many substrings in the sentence and you will also need to count and display each of the word's length.

Sample Input	Sample Output

`Welcome To Java Programming Class`	`Substrings length:5`
	`Str[0]:Welcome Length:7`
	`Str[1]:To Length:2`
	`Str[2]:Java Length:4`
	`Str[3]:Programming Length:11`
	`Str[4]:Class Length:5`
	`Ï`

`Selamat;Hari-Raya Aidilfitri`	`Substrings length:4`
	`Str[0]:Selamat Length:7`
	`Str[1]:Hari Length:4`
	`Str[2]:Raya Length:4`
	`Str[3]:Aidilfitri Length:10`

`Programming-Is-Fun!`	`Substrings length:3`
	`Str[0]:Programming Length:11`
	`Str[1]:Is Length:2`
	`Str[2]:Fun! Length:4`

Problem setter: MAHFUDZAH OTHMAN

PROBLEM C: FIND THE WINNER

The 2016 SUKMA Games was held in Sarawak from 23 July – 31 July 2016. During the closing ceremony, the medal tally and team ranking was shown to the audience. Teams were ranked based on the number of gold medals obtained by each team. The team in the first rank is the winner of SUKMA 2016.

Input

The first line of the input contains the number of teams participated in SUKMA 2016. Following the first line are the test cases. Each line in a test case contains the team name and 3 integer values that represent the gold, silver and bronze medal respectively. Assume that each team obtained different number of gold medal.

The input must be read from standard input.

Output

The output of the program should print the team ranking and the winner of SUKMA 2016.

The output must be written to standard output.

Sample Input	Sample Output
5 Kedah 8 4 5 Perak 9 7 7 Johor 12 4 8 Perlis 10 5 6 Melaka 7 4 5	Team Ranking: Johor 12 4 8 Perlis 10 5 6 Perak 9 7 7 Kedah 8 4 5 Melaka 7 4 5 Winner: Team Johor

Problem setter: Hawa Mohd Ekhsan

PROBLEM D: PAIR ISOGRAMS

A word is considered to be "pair isogram" if each letter in the word appears exactly twice (not less, not more) in the word. For example, the word "teammate" is pair isogram since each letter in the word appears exactly twice (not less, not more), but the word "dad" is not since the letter "a" of the word doesn't appear twice.

Your task is to write a program that reads a word and then determines whether or not it is pair isogram.

Input
The first line of the input is a positive integer N (1 ≤N ≤ 10), indicating the number or words to be processed. The next line of input contains a word that has to be determined whether **pair isogram** or **not pair isogram** . Assume each input word will contain only lowercase letter (no other characters) and will be at least one letter.

The input must be read from standard input.

Output
For each word, print input word along with a message indicating whether or not it is a pair isogram. In between input word and message, output "---". Follow the format illustrated in Sample Output.

The output must be written to standard output.

Sample Input	Sample Output
3 teammate ali dood	teammate --- pair isogram ali --- not pair isogram dood --- pair isogram

Problem setter: MOHD NIZAM OSMAN

PROBLEM E: ENCRYPTION

Encryption is the conversion of electronic data into another form, called cipher text, which cannot be easily understood by anyone except authorized parties. Your task is to write a program that is able to encrypt any given plain text into cipher text using the following encryption algorithm:

(i) All characters except space (“ “) in the plain text are changed to its next TWO (2) character in the ASCII character table. Example: Letter K will be changed to letter M.

(ii) Spaces are changed to another symbol based on its occurrences. For example, its first occurrence will be replaced with the symbol next to it in the table and its second occurrence will be replaced by its next two symbol.

Input
Plain text.

Output
Cipher text.

Sample Input	Sample Output
`This is C-PROM 2016.`	`Vjku!ku"E/RTQO#42380`
`Hie, there. I am competing in a programming contest.`	`Jkg.!vjgtg0"K#co$eqorgvkpi%kp&c'rtqitcoo kpi(eqpvguv0`

Problem setter: NORA YANTI CHE JAN

PROBLEM F: TITLE CASE

Title gives a brief idea on overall content of particular subject matter. While name represents an identity of a person. When writing a title or name of a person, usually first character of each word is capitalized.

Input
First line of input contains an integer N ($1 \leq N \leq 10$) to indicate number of test cases. Following the first line are the test cases. Each line of a test case contains a string of words W($W > 0$).

The input must be read from standard input.

Output
The output of the program should display the string where the first character of each word is capitalized.

The output must be written to standard output.

Sample Input	Sample Output
5 a b c aa bbb cccc linus torvalds welcome to copscom 2016!!! muhammad ibn musa al Khwarizmi	A B C Aa Bbb Cccc Linus Torvalds Welcome To Copscom 2016!!! Muhammad Ibn Musa Al Khwarizmi

Problem setter: MUHAMMAD NABIL FIKRI JAMALUDDIN

PROBLEM G: SIMPLE AVERAGE AND MEDIAN

Create a program that allows a user to enter 1 to 10 integer numbers on a single line of input (you may assume that the input contains no more than 10 numbers and no less than 1 number). Your task is to compute the average and median of the list and display the result

Input
The first line of the input contains an integer N (1 ≤ N ≤ 5), the number of test cases. Following the first line is the total number to be entered.The next line represents the integer numbers.

The input must be read from standard input.

Output
The output of the program should displaythe average and median asthe example below:

The output must be written to standard output.

Sample Input	Sample Output
3 5 1 2 5 10 15 5 10 20 30 40 50 3 3 4 5	Average:6.6 Median:5 Average:30 Median:30 Average:4 Median:4

Problem setter: MOHD NIZAM OSMAN

PROBLEM H: LEXICOGRAPHIC ORDERING

When comparing two strings using lexicographic (dictionary) ordering, corresponding letters are compared until one of the string ends or the first difference is encountered.

If one of the strings ends, the longer string is considered the later one. If a character mismatch is found, compare the characters to determine which string comes later in the sequence.

Input

The first line of the input contains the number of strings to be tested. Following the first line are the test cases. Each line in a test case contains a string with maximum 30 characters.

For example,

3
Tomato
Tony
Tom

The input must be read from standard input.

Output

The output of the program should print the strings in lexicographic order.

The output must be written to standard output.

Sample Input	Sample Output
4 compute comparison compete computer	comparison compete compute computer

Problem setter: HAWA MOHD EKHSAN

PROBLEM I: SERIES OF NUMBERS

Write a program that is able to generate **FIVE (5)** strings which are greater that the input string. The program will initially increment the numbers in the given string but if the number is greater than 9999, it will increment the last alphabet in the string and restart the number to 1.

Input:
A string that starts with one or more alphabets followed by one or more digits.

Output:
Five strings with numbers or alphabets greater than the input string.

Sample of input	Sample of output
M1234	M1235 M1236 M1237 M1238 M1239
RM9997	RM9998 RM9999 RN1 RN2 RN3
WPA9998	WPA9999 WPB1 WPB2 WPB3 WPB4

Problem setter: NORA YANTI CHE JAN

PROBLEM J: ADDING HEXADECIMAL NUMBER

The hexadecimal system (base 16) with its 16 digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) allows for more compact representation of the number. One of the many nice things that can be done with hexadecimal numbers is adding them. For example:

A + 5 = F
AA + 11 = BB
AA + FF = 1A9

Your task is to add two hexadecimal numbers.

Input
The first line of the input contains an integer N (1 ≤N ≤ 10) indicating the number of pairs of hexadecimal numbers. The next line of input will contain two hexadecimal numbers. The hexadecimal numbers will be positive, of arbitrary length, be in uppercase, and will have at most 256 digits.

The input must be read from standard input.

Output
The output will print the summation of the hexadecimal numbers indicating the number for the corresponding addition question.

The output must be written to standard output.

Sample Input	Sample Output
3 8 3 59 A7 4FE 20	 B 100 51E

Problem setter: MOHD NIZAM OSMAN

Real Competition
PROBLEM A: MULTIPLICATION TABLE *(C++)*

```
#include<iostream>
usingnamespacestd;

int main()
{
intnum;
int count =1;
cin>>num;
if (num<1 || num>12)
cout<<"Invalid";
else{
while(count <= 12)
   {
cout<<count <<"*"<<num<<"="<<count*num<<endl;
      count++;}
   }
      }
```

Real Competition
PROBLEM B: SPLIT & COUNT *(C++)*

```
import java.util.*;
public class StringSplitTest{
public static void main(String args[]){
String s;// = "Welcome To Java Programming Class";
Scanner m = new Scanner(System.in).useDelimiter("//s");
s = m.nextLine();
String[] strs = s.split("[,\\- \\s \\;]");

System.out.println("Substrings length:"+strs.length);
for ( int i=0; i < strs.length; i++){
int length = strs[i].length();
System.out.println("Str["+i+"]:"+strs[i]+" Length:"+length);
        }

    }
  }
```

Real Competition
PROBLEM C: FIND THE WINNER *(C++)*

```
#include <iostream>
using namespace std;

struct sukma
{
      char teamName[30];
      int gold, silver, bronze;
};

int main(void)
{
      int noOfTeam;
      cin>>noOfTeam;

      sukma team[noOfTeam];
      sukma temp,winner;

      for (int i=0; i<noOfTeam; i++)
      {
            cin>>team[i].teamName>>team[i].gold>>team[i].silver>>
                  team[i].bronze;
      }

      for (int j=0; j<noOfTeam; j++)
      {
            for (int i=(j+1); i<noOfTeam; i++)
            {
                        if (team[j].gold < team[i].gold)
                        {
                              temp = team[i];
                              team[i] = team[j];
                              team[j]=temp;
                        }
          }
      }
      winner= team[0];
      cout<<"\nTeam Ranking:\n";

      for (int i=0; i<noOfTeam; i++)
      {
            cout<<team[i].teamName<<"\t"<<team[i].gold<<"\t"
            <<team[i].silver<<"\t"<<team[i].bronze<<"\n";
      }
      cout<<"\nWinner: Team " <<winner.teamName;
}
```

Real Competition
PROBLEM D: PAIR ISOGRAMS *(Java)*

```
import java.util.*;

public class Isogram{
      public static void main(String[] args) {
      Scanner scan = new Scanner(System.in);
      Scanner scanN = new Scanner(System.in);


         int N = scanN.nextInt();
         String[] data = new String[N];

         for (int i=0; i<N; i++)
            data[i] = scan.nextLine();

         System.out.println("\n\n");
         for(int i=0; i<N; i++)
         {
             int[] cnt = new int[26];

             StringTokenizer st = new StringTokenizer(data[i]);
             String token = st.nextToken();

             for(int b=0; b<token.length();b++){
      cnt[token.charAt(b)-'a']++;
             }

             boolean legit = true;

             for(int b=0; b<26;b++){
      if(cnt[b] !=0 && cnt[b] != 2)
            legit = false;
             }

             if (legit)
      System.out.println(token+ "--- pair isogram");
             else
      System.out.println(token+ "--- not pair isogram");
         }
      }
}
```

Real Competition
PROBLEM E: ENCRYPTION *(Java)*

```
import java.util.*;

public class Question1
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);

         System.out.print("Please enter your message: ");
        String message = in.nextLine();

        char currAlphabet;
        String eMessage = " ";
         intcountSpace = 0;
        for(int index = 0; index <message.length(); index++)
        {
           currAlphabet = message.charAt(index);
           if (currAlphabet == ' ')
           {
           countSpace++;
           currAlphabet = (char)((int) currAlphabet + countSpace);
           }
           else
           {
               currAlphabet = (char)((int) currAlphabet + 2);
           }

System.out.print(currAlphabet);
            //eMessage.charAt(index) = currAlphabet;
        }
    }
}
```

Real Competition
PROBLEM F: TITLE CASE *(Java)*

```java
import java.util.Scanner;
import java.util.StringTokenizer;

public class Capitalization {
	public static void main(String[] args) {

		Scanner sc = new Scanner(System.in);

		int N = Integer.parseInt(sc.nextLine());
		String testCase[] =  new String[N];

		for(int i = 0; i<N; i++){
			testCase[i] = sc.nextLine();
		}

		for(inti = 0; i<N; i++){
			String newString = "";
			charc = ' ';
			StringTokenizerst = newStringTokenizer(testCase[i], " ");
			String token = "";
			while(st.hasMoreTokens()){
				token = st.nextToken();
				c = Character.toUpperCase(token.charAt(0));
				token = token.substring(1, token.length());
				token = c + token;
				newString += token + " ";
			}
			System.out.println(newString);
		}
		sc.close();
	}
}
```

Real Competition
PROBLEM G: SIMPLE AVERAGE AND MEDIAN

```
#include<iostream>
using namespace std;

int main()
{
      int N, indMed, totNum, sum;
      int number[10];
      double avg[5], med[5];

      cin >>N;
      for(int i=0; i<N; i++){
            indMed = 0;
            sum=0;
            cin >> totNum;
            for(int j=0; j<totNum; j++)
                  cin>> number[j];

            for(int j=0; j<totNum; j++){
                  sum += number[j];
            }
            avg[i] = (sum+0.0)/totNum;
            indMed = totNum/2;
            med[i] = number[indMed];
      }

      for(int i=0; i<N; i++){
            cout << "Average: "<<(double)avg[i]<<"  Median: " <<med[i]
<< endl;
      }
}
```

Real Competition
PROBLEM H: LEXICOGRAPHIC ORDERING *(C++)*

```
#include <iostream.h>
#include<string.h>

void main()
{
      int i,j,numWords;
      char string[10][30], temp[30];
      cin>>numWords;

      for (i=0; i<numWords; i++)
      {
            cin>>string[i];
      }

      for (i=0; i<numWords; i++)
      {
            for (j=(i+1); j<numWords; j++)
            {
                        if(strcmp(string[i],string[j])>0)
                        {
                              strcpy(temp,string[i]);
                              strcpy(string[i], string[j]);
                              strcpy(string[j], temp);
                        }
            }
      }
      cout<<endl;
      for (i=0; i<numWords; i++)
      {
            cout<<string[i]<<endl;
      }

}
```

Real Competition
PROBLEM I: SERIES OF NUMBERS *(Java)*

```
import java.util.*;
public class question1
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        String regNo = in.next();
 //generating the next 5 registration number
int lengthRegNo = 0;
lengthRegNo = regNo.length();
int numIndex = 0;

for (int index = 0; index <lengthRegNo; index++)
        {
            if (Character.isDigit(regNo.charAt(index)))
            {
           numIndex = index;
                break;
            }
        }
int lastAlphabetIndex = numIndex-1; //index of the last alphabet
char lastAlphabet = regNo.charAt(lastAlphabetIndex);
int lastAlphabetASCII = (int) lastAlphabet;
int newLastAlphabet;
int numberOnly = Integer.parseInt(regNo.substring(numIndex));
for (int counter = 0; counter < 5; counter++)
        {
         if (numberOnly == 9999)
         {
newLastAlphabet = lastAlphabetASCII + 1;
lastAlphabet = (char) newLastAlphabet;
numberOnly = 1;
         }
         else
numberOnly++;
System.out.println(regNo.substring(0, lastAlphabetIndex) +
lastAlphabet + numberOnly);
        }   }
```

Real Competition
PROBLEM J: ADDING HEXADECIMAL NUMBER *(Java)*

```
import java.util.*;
import java.io.*;
public class Hexadecimal {
      public static void main(String[] args) {
      Scanner scan = new Scanner(System.in);
      Scanner scanN = new Scanner(System.in);

         int N = scanN.nextInt();
         String[] data = new String[N];

         for (int i=0; i<N; i++)
            data[i] = scan.nextLine();

         System.out.println("\n\n");
         for(int i=0; i<N; i++)
         {
             StringTokenizer st = new StringTokenizer(data[i]);
             String strNum1 = st.nextToken().toUpperCase();
             String strNum2 = st.nextToken().toUpperCase();

             String sum = findSum(strNum1,strNum2);
             System.out.println(sum);
         }
      }
public static String findSum(String n1, String n2)
      {
            int carry=0, sum=0;
            String result=new String();
            int i;

            if(n1.length()>n2.length()){
                  while(n1.length()>n2.length())
                        n2 = "0"+n2;
            }
            else if(n2.length()>n1.length()){
                  while(n2.length()>n1.length())
                        n1="0"+n2;}
      for(i=n1.length()-1;i>=0;i--){
      int d1=findDecimal(n1.charAt(i));
      int d2=findDecimal(n2.charAt(i));
      sum=d1+d2+carry;
      String val=add(sum);
      result=val.charAt(val.length()-1)+result;
```

```
  if(val.length()==2)
        carry=Integer.parseInt(val.substring(0, 1));
              else
                    carry=0;
        }
        if(carry!=0)
              result=carry+result;
        return result;
  }
  public static int findDecimal(char ch)
  {
        char h[]={'0', '1', '2', '3', '4', '5', '6', '7', '8',
                    '9', 'A', 'B', 'C', 'D', 'E', 'F'};
        int d[]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
                    14, 15};
        for(int i=0;i<h.length;i++)
              if(ch==h[i])
                    return d[i];
        return 0;
  }
  public static char findHexa(int num)
  {
        char h[]={'0', '1', '2', '3', '4', '5', '6', '7', '8',
                    '9', 'A', 'B', 'C', 'D', 'E', 'F'};
        int d[]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
                    14, 15};

        for(int i=0;i<h.length;i++)
              if(num==d[i])
                    return h[i];
        return ' ';
  }
  public static String add(int sum)
  {
        String str=new String();

        while(sum>=10){
              int rem=sum%16;

              if(rem>=10)
                    str=findHexa(rem)+str;
              else
                    str=rem+str;
              sum=sum/16;
        }
        if(sum!=0)
              str=sum+str;
        return str;
  }
}
```