Talent Rank

ZOHO L2 MAY 29 SLOT 2 REPORT

L

Loksundar

loksundar000@gmail.com

May 29, 2021

✓ Qualified

239m (Time Taken)

5/6 (Problems Solved)

70.0 (Scored out of 100)

Consolidated Marks - Section Wise

Section	Score
Kongu drive --- Programming	**70.0**/100.0
- Convert Number to English (10 marks)	**0.1**/0.1
- String Rotation (10 marks)	**0.1**/0.1
- Array Sorting (10 marks)	**0.1**/0.1
- Value of string (10 marks)	**0.1**/0.1
- Shortest time duration (30 marks)	**0.3**/0.3
- Remove the common fragment in given strings (30 marks)	**0.0**/0.3
Total Score	**70.0**/100.0

Kongu drive --- Programming

Convert Number to English (10 marks)

Score: **0.1** / 0.1

Accepted

Program to convert a given number to words is discussed here. For example, if 1234 is given as input, the output should be one thousand two hundred and thirty four".

Input:
Input consist of an one integer.
Output:
Refer to the sample input and output for formatting specifications.

Sample Input:
438237764

Sample Output:

forty three crore eighty two lakh thirty seven thousand seven hundred and sixty four

Solution

```
import java.util.Scanner;
class Main {
  static String one[] = {"","one ","two ","three ","four ","five ","six ","seven ","eight ","nine ","ten ","eleven ","twelve ","thirteen ","fourteen ","fifteen ","sixteen ","seventeen ","eighteen ","nineteen "};
  static String ten[] = {"","","twenty ","thirty ","forty ","fifty ","sixty ","seventy ","eighty ","ninety "};
  static String ntow(int n, String s){
    String str="";
    if(n>19){
      str+=ten[n/10]+one[n%10];
    }
    else{
      str+= one[n];
    }
    if(n!=0){
      str+=s;
    }
    return str;
  }
  static String convtow(long n){
    String out="";
    out += ntow((int)(n/10000000),"crore ");
    out += ntow((int)((n/100000)%100),"lakh ");
    out += ntow((int)((n/1000)%100),"thousand ");
    out += ntow((int)((n/100)%10),"hundred ");
    if(n>100 && n%100>0) out +="and ";
    out += ntow((int)(n%100),"");
    return out;
  }
  public static void main(String args[]) {
    Scanner sc=new Scanner(System.in);

    long n=sc.nextLong();
    System.out.printf(convtow(n));


  }
}
```

Test Cases

- **Testcase - 1**
  Purpose - General
- **Testcase - 2**
  Purpose - General
- **Testcase - 3**
  Purpose - General
- **Testcase - 4**
  Purpose - General
- **Testcase - 5**
  Purpose - General

String Rotation (10 marks)

Score: **0.1** / 0.1

Accepted

How do you check if two strings are a rotation of each other?

Input:
Input consist of two strings.

Output:
Print “Yes” if two strings are a rotation of each other, otherwise print “No”.

Sample Input 1:
"Hello from here"
"reHello from he"
Sample Output 1:
Yes

Sample Input 2:
"Hello from here"
"erHello from he"
Sample Output 2:
No

Solution

```java
import java.util.Scanner;
class Main {
  static boolean rotateWord(String str1,String str2){
    return (str1.length()==str2.length()) && ((str1+str1).contains(str2));
  }


  public static void main(String args[]) {
    Scanner scan = new Scanner(System.in);
    String str1 = scan.nextLine();
    String str2 = scan.nextLine();
    if(rotateWord(str1,str2)){
      System.out.println("Yes");
    }
    else{
      System.out.println("No");
    }


  }
}
```

Test Cases

✓ Testcase - 1
Purpose - General

✓ Testcase - 2
Purpose - General

✓ Testcase - 3
Purpose - General

✓ Testcase - 4
Purpose - General

✓ Testcase - 5
Purpose - General

✓ Testcase - 6
Purpose - General

✓ Testcase - 7
Purpose - General

Array Sorting (10 marks)

Score: **0.1 / 0.1** ✓ Accepted

Sorting Array

Write a program to sort the array of objects colored red, white or blue in the order of red, white and blue in which the objects of same color are adjacent.
Here, we will use the integers 0, 1, and 2 to represent the color red, white, and blue respectively.

Input Format:
The first line of the input consists of an integer n, which corresponds to the number of elements in an array.
Next line of the input consists of n space-separated integers, which corresponds to the elements of an array. Assume that the array elements will be 0, 1 and 2, which represents red, white and blue, respectively.

Output Format:
Output line displays the sorted order the array according to the color separated by a space.
Refer to the sample input and output for formatting specifications.

Sample Input 1:
7
0 1 2 1 0 2 1
Sample Output 1:
0 0 1 1 1 2 2

Sample Input 2:
5
0 1 0 1 0
Sample Output 2:
0 0 0 1 1

Solution

```
#include<stdio.h>
int main()
{
  int i,j,a,n,number[50];
  scanf("%d",&n);
  for(i=0;i<n;++i) scanf("%d",&number[i]);
  for(i=0;i<n;++i){
    for(j=i+1;j<n;++j){
      if(number[i]>number[j]){
        a=number[i];
        number[i]=number[j];
        number[j]=a;
      }
    }
  }
  for(i=0;i<n;++i) printf("%d ",number[i]);

  return 0;
}
```

Test Cases

✓ **Testcase - 1**
Purpose - General

✓ **Testcase - 2**
Purpose - General

✓ **Testcase - 3**
Purpose - General

✓ **Testcase - 4**
Purpose - General

✓ **Testcase - 5**
Purpose - General

✓ **Testcase - 6**
Purpose - General

✓ **Testcase - 7**
Purpose - General

✓ **Testcase - 8**
Purpose - General

Value of string (10 marks)

Score: **0.1 / 0.1** ✓ Accepted

You are given a String **s** consisting of lower case letters. The letters 'a' to 'z' are assigned to values 1 to 26, respectively. The value of **s** is defined to be, the sum of the following: (number of times the letter in s[i] is repeated) X (value of the letter).

Given the string, compute and return the value of the string as an integer.

Example

1.
   1. Input : "babca"
      Output : 9
      Explanation : The value of this string is 2*2 + 2*1 + 1*3 = 9.

   2. Input : "zz"
      Output : 52
      Explanation : The value of this string is 2*26 = 52.

   3. Input : "y"
      Output : 25
      Explanation : The value of this string is 1*25 = 25.

Input Format:
Input consists of one string.
Note: Input Strings consists of lower case letters.

Output Format:
Refer to the sample input and output for formatting specifications.

Sample Input 1:
abczzzzzzzabcdef
Sample Output 1:
209

Sample Input 2:
yuidnofkggdfgdhd
Sample Output 2:
152

Solution

```
using namespace std;
#include<iostream>
#include<bits/stdc++.h>
int main()
{
  string s;
  cin>>s;
  map<char,int>m;
  for(int i=0;s[i];i++)
  {
    if(m.count(s[i])){
      m[s[i]]++;
    }
    else{
      m[s[i]]+=1;
    }
  }
  int sum=0;
  for(auto it:m){
    int ascii_val=it.first;
    ascii_val -=96;
    sum = sum +(ascii_val*it.second);
  }
  cout<<sum;


  return 0;
}
```

Test Cases

Testcase - 1
Purpose - General

Testcase - 2
Purpose - General

Testcase - 3
Purpose - General

Testcase - 4
Purpose - General

Testcase - 5
Purpose - General

Testcase - 6
Purpose - General

Testcase - 7
Purpose - General

Shortest time duration (30 marks)

Score: 0.3 / 0.3 Accepted

Print the shortest duration between the time values given in the input array. The time values are given in the format (hour:minute:second). The time values in the array are in no particular order.

Example:
Input : {12:34:55,1:12:13,8:12:15}
output :
4:22:40
Explanation :
12:34:55 - 8:12:15 = 4:22:40
12:34:55 - 1:12:13 = 11:22:42
8:12:15 – 1:12:13 = 7:0:02
smallest is 4:22:40

Input Format:
The first line of the input consists of an integer, N that corresponds to the number of durations in the input array.
The next N lines of the input corresponds to the durations in a given array. Assume that the duration are given in the format: hours:minutes:seconds

Output Format:
Output is to display the shortest time duration (in hours:minutes:seconds) of all values in a given array
Refer to the sample input and output for formatting specifications.

Sample Input 1:
3
12:34:55
1:12:13
8:12:15
Sample Output 1:
4:22:40

Sample Input 2:
5
12:13:50
16:30:00
01:00:15
06:30:45
09:15:15
Sample Output 2:
2:44:30

Solution

```
import java.util.Scanner;
import java.util.Arrays;
import java.io.IOException;
class Main {
  public static int parseTime(String str){
    int h,m,s;
    String units[]=str.split(":");
    h = Integer.parseInt(units[0]);
    m = Integer.parseInt(units[1]);
    s = Integer.parseInt(units[2]);
    return (h*3600 + m*60 +s);
  }
  public static void main(String args[]) throws IOException{

    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    sc.nextLine();
    int i,j;
    String[] arr = new String[n];
    int t[] = new int[n];
   for(i=0;i<n;i++){   arr[i] =sc.nextLine();
      t[i] = parseTime(arr[i]);
   }
    Arrays.sort(t);

    int diff[] = new int[n-1];
    for(i=0;i<n-1;i++){
      diff[i]=t[i+1]-t[i];
    }
    Arrays.sort(diff);
    int h = diff[0]/3600;
    int m = (diff[0]%3600)/60;
    int s = (diff[0]%3600)%60;
    System.out.printf("%s:%s:%s",h,m,s);
  }
}
```

Test Cases

- ✓ **Testcase - 1**
  Purpose - General
- ✓ **Testcase - 2**
  Purpose - General
- ✓ **Testcase - 3**
  Purpose - General
- ✓ **Testcase - 4**
  Purpose - General
- ✓ **Testcase - 5**
  Purpose - General
- ✓ **Testcase - 6**
  Purpose - General
- ✓ **Testcase - 7**
  Purpose - General
- ✓ **Testcase - 8**
  Purpose - General

Remove the common fragment in given strings (30 marks)

Score: **0.0 / 0.3**

Reqmts Mismatch

Remove a fragment

Write a program to find a fragment that occurs in all strings, where a fragment is 3 consecutive words.

Note: If you are writing this program in JAVA, don't use built-in functions like split(), indexOf(), replace(), substring(), etc present in String Class. Do not hard code the output.

Example:
Given three strings like:
    S1 = "Every morning I want to do exercise regularly"
    S2 = "Every morning I want to do meditation without fail"
    S3 = "It is important that I want to be happy always"

Then the :
    Common fragment = “I want to”

Explanation:
    “I want to” is the common fragment (3 continuos words) found in all three sentences.

Input Format:
First line of the input is an integer n, which corresponds to the number of strings.
Second line of the Input consists of n strings line by line

Output Format:
Display the common fragment.
Refer to the sample input and output for formatting specifications.

Sample Input:
3
Every morning I want to do exercise regularly
Every morning I want to do meditation without fail
It is important that I want to be happy always
Sample Output:
I want to

Solution

```
import java.util.Scanner;
import java.io.*;
import java.lang.String;
class Main {
  public static String[] splt(String str){
    int i,j,space=0;
    String c=" ";
    char ch = c.charAt(0);
    for(i=0;i<str.length();i++){
      if(str.charAt(i)==ch) space++;
    }
    String arr[] = new String[space+1];
    String nstr ="";
    for(i=0,j=0;i<str.length();i++){
      if(str.charAt(i)==ch){
        arr[j]=nstr;
        j++;
        nstr="";
      }
      else{
        nstr = nstr+str.charAt(i);
      }
      arr[j]=nstr;
    }
    return arr;


  }
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    String s1=sc.nextLine();
    String s2=sc.nextLine();
    String s3=sc.nextLine();
    String []a = splt(s1);
    for(int i=0;i+2<a.length;i++){
      String b=a[i]+""+a[i+1]+""+a[i+2];
      if(s1.indexOf(b)!=-1 && s2.indexOf(b)!=-1 && s3.indexOf(b)!=-1){
        System.out.print(b);
        break;
      }
    }


  }
}
```

Test Cases