Rajalakshmi Engineering College

Name: Elamvazhuthi MS
Email: 240701133@rajalakshmi.edu.in
Roll no: 240701133
Phone: 7010282287
Branch: REC
Department: CSE - Section 4
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 9_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 34.5

Section 1 : Coding

1. Problem Statement

Aarav is developing a music playlist application where users can manage their favorite songs. He wants to implement a feature that allows users to reorder the playlist by moving a song from one position to another.

You need to implement a function that performs the following operations using a LinkedList:

Add songs to the playlist in the given order.Move a song from a specified position to another position in the playlist.Print the final playlist after all operations.

Input Format

The first line of the input consists of an integer n representing the number of songs.

240701133

The next n lines, each containing a string representing a song name.

After the songs are given the next line contains an integer m, the number of move operations.

The next m lines, each containing two integers x and y representing the move operation where the song at position x (0-based index) should be moved to position y.

Output Format

The output prints the final playlist, each song on a new line.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
SongA
SongB
SongC
SongD
SongE
2
2 4
0 3
Output: SongB
SongD
SongE
SongA
SongC

Answer

```
  import java.util.*;

class PlaylistManager {
   public static void main(String[] args) {
      Scanner s = new Scanner(System.in);
      int n = s.nextInt();
```

240701133

```
        s.nextLine();
        LinkedList<String> playlist = new LinkedList<>();
        for (int i = 0; i < n; i++) {
            playlist.add(s.nextLine());
        }
        int m = s.nextInt();
        for (int i = 0; i < m; i++) {
            int x = s.nextInt();
            int y = s.nextInt();
            String song = playlist.remove(x);
            playlist.add(y, song);
        }
        for (String song : playlist) {
            System.out.println(song);
        }
        s.close();
    }
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Raman, a computer science teacher, is responsible for registering students for his programming class. To streamline the registration process, he wants to develop a program that stores students' names and allows him to retrieve a student's name based on their index in the list.

Raman has decided to use an ArrayList to store the names of students, as it provides efficient dynamic resizing and indexing.

Write a program that enables Raman to input the names of students and fetch a student's name using the specified index. If the entered index is invalid, the program should return an appropriate message.

Input Format

The first line of input consists of an integer n, representing the number of

240701133

students to register.

The next n lines of input consist of the names of each student, one by one.

The last line of input is an integer, representing the index (0-indexed) of the element to retrieve.

Output Format

If the index is valid (within the bounds of the ArrayList), print "Element at index [index]: " followed by the element (student name as string).

If the index is invalid, print "Invalid index".

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 5
Alice
Bob
Ankit
Alice
Prajit
2
Output: Element at index 2: Ankit
```

Answer

```
// You are using Java
import java.util.*;
import java.util.ArrayList;
class d{
   public static void main(String[] args){
      Scanner s=new Scanner(System.in);
      int n=s.nextInt();
      ArrayList<String> a=new ArrayList<>();
      for(int i=0;i<n;i++){
         String f=s.next();
         a.add(f);
      }
      int d=s.nextInt();
```

240701133

```
        if(a.size()<=d){
          System.out.printf("Invalid index");

        }
        else{
          System.out.print("Element at index "+d+": "+a.get(d));
        }
    }
}
```

Status : *Partially correct* ***Marks : 8.5/10***

3. Problem Statement

A teacher is filtering a list of words provided by students. Some words contain too many vowels, making them difficult for a spelling competition. The teacher decides to remove all words that contain more than two vowels.

Help the teacher to implement it using ArrayList.

Input Format

The first line contains an integer N, representing the number of words in the list.

The next N lines contain a string representing the words (one per line).

Output Format

The output consists of words that contain two or less than two vowels, printed in the same order they appeared in the input. Each word is printed on a new line.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1
sri
Output: sri

240701133

Answer

```
import java.util.ArrayList;
import java.util.Scanner;

// You are using Java
class VowelFilter {

  static void filterWords(int n,Scanner sc){
      ArrayList<String>a=new ArrayList<>();
      for(int i=0;i<n;i++){
         String S=sc.next();
         String[] k=S.split("");
         int c=0;
         for(int l=0;l<k.length;l++){
            if(k[l].equals("a")||k[l].equals("A")){
               c++;
            }
            else if(k[l].equals("E")||k[l].equals("e")){
               c++;
            }
            else if(k[l].equals("i")||k[l].equals("I")){
               c++;
            }
            else if(k[l].equals("o")||k[l].equals("O")){
               c++;
            }
            else if(k[l].equals("u")||k[l].equals("U")){
               c++;
            }
         }
         if(c<=2){
         a.add(S);}
      }
      for(int i=0;i<a.size();i++){
         System.out.println(a.get(i));
      }
   }
}

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
```

240701133

```
        sc.nextLine();
        VowelFilter.filterWords(n, sc);
        sc.close();
    }
}
```

Status : Partially correct ***Marks : 8.5/10***

4. Problem Statement

Rahul, a stock trader, wants to analyze the stock prices of a company over several days. For each day, he wants to determine the stock span, which is the number of consecutive days (including the current day) where the stock price is less than or equal to the price on that day.

The stock span helps him understand how long a stock has been continuously increasing or staying the same. You need to help Rahul by computing the stock span for each day using a Stack data structure efficiently.

Example:

Input:

7

100 80 60 70 60 75 85

Output:

1 1 1 2 1 4 6

Explanation:

For each day:

Day 1: Price = 100   Span = 1 (Only this day)Day 2: Price = 80   Span = 1 (Only this day)Day 3: Price = 60   Span = 1 (Only this day)Day 4: Price = 70   Span = 2 (Includes today and previous day)Day 5: Price = 60   Span = 1 (Only this day)Day 6: Price = 75   Span = 4 (Includes today and previous three days)Day 7: Price = 85   Span = 6 (Includes today and previous five days)

240701133

Input Format

The first line contains an integer n, the number of days.

The second line contains n space-separated integers prices[i], where prices[i] represents the stock price on the i-th day.

Output Format

The output prints n space-separated integers representing the stock span for each day.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7
100 80 60 70 60 75 85

Output: 1 1 1 2 1 4 6

Answer

```
// You are using Java
import java.util.*;
class ss{
   public static void main(String[] args){
      Scanner s=new Scanner(System.in);
      int n=s.nextInt();
      Stack<Integer> a=new Stack<>();
      for(int i=0;i<n;i++){
         int h=s.nextInt();
         a.add(h);
      }
      Stack<Integer>st=new Stack<>();
      st.add(1);
      //int c=1;
      for(int i=1;i<n;i++){

            if(a.get(i-1)>a.get(i)){
               st.add(1);
              // c++;
            }
```

240701133

```
            else{
              int k=a.indexOf(a.get(i));
              int c=0;
               for(int j=0;j<=k;j++){
                  // for(int h=j;h<k;h++){
                   if(a.get(j)<=a.get(k)){
                       c++;
                   }
               //}

            }
            st.add(c);

            }



        }
        for(int i=0;i<st.size();i++){
            System.out.print(st.get(i)+" ");
        }
    }
}
```

Status : Partially correct ***Marks : 7.5/10***

240701133