DAY 9:

SESSION 1:

Problem 1:

Stock Buy & Sell Once - Maximum Profit

ID:10852 Solved By 819 Users

The program must accept the stock prices on **N** days as the input. A person can buy a stock on a particular day and he can sell it once on any other given day. He can not buy and sell on the same day. The program must print the maximum possible profit **P** that can be obtained by buying and selling 1 stock once as the output.

Boundary Condition(s):
2 <= N <= 10^5
1 <= Each integer value <= 10^5

Input Format:
The first line contains N.
The second line contains N integers separated by a space.

Output Format:
The first line contains P.

Example Input/Output 1:
Input:
7
50 100 40 60 70 50 80

Output:
50

Explanation:
Here N = **7**.
The stock price on the **1st** day is **50** and the stock price on the **2nd** day is **100**.
On buying the stock on the 1st day and selling it on the 2nd day can earn the maximum profit **50** (100 - 50 = 50).
Hence the output is 50

Example Input/Output 2:
Input:
10
15 10 60 70 45 5 70 30 100 90

Output:
95

Max Execution Time Limit: 500 millisecs

Code:

```
import java.util.*;
public class stockBuySellOnceMaxProfit {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in = new Scanner(System.in);
        int N=in.nextInt();
        int[] prices=new int[N];
        for(int index=0;index<N;index++){
            prices[index]=in.nextInt();
        }
        int minPrice=prices[0],maxProfit=0;
```

```
        for(int index=1;index<N;index++){
            if(prices[index]<minPrice){
                minPrice=prices[index];
            }
            else{
                int profit = prices[index]-minPrice;
                if(profit>maxProfit){
                    maxProfit=profit;
                }
            }
        }
        System.out.println(maxProfit);
    }
}
```

Problem 2:

Stock Buy & Sell Multiple Times - Maximum Profit

ID:11119 Solved By 811 Users

The program must accept the stock prices on **N** days as the input. A person can buy a stock on a particular day and he can sell it once on any other given day. He can not buy and sell on the same day. The program must print the maximum possible profit **P** that can be obtained by buying and selling the stocks **multiple times** as the output.
Note: The person can buy only one stock at a time and the person can buy another stock only after selling it.

Boundary Condition(s):
2 <= N <= 10^5
1 <= Each integer value <= 10^5

Input Format:
The first line contains N.
The second line contains N integers separated by a space.

Output Format:
The first line contains P.

Example Input/Output 1:
Input:
10
5 8 10 12 9 6 14 21 15 10

Output:
22

Explanation:
Here N = **7**.
The maximum profit is obtained by buying & selling the stocks in the following ways.
On buying the stock on the 1st day and selling it on the 4th day can earn the profit **7** (12 - 5 = 7).
On buying the stock on the 6th day and selling it on the 8th day can earn the profit **15** (21 - 6 = 15).
So the total profit is **22** (7 + 15).

Example Input/Output 2:
Input:
9
1 2 3 1 20 30 10 5 6

Output:
32

Max Execution Time Limit: 500 millisecs

Code:

```java
import java.util.*;
public class stockBuySellMultipleTimesMaxProfit {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in = new Scanner(System.in);
        int N=in.nextInt();
        int[] prices = new int[N];
        for(int index=0;index<N;index++){
            prices[index]=in.nextInt();
        }
        int profit=0;
        for(int index=1;index<N;index++){
            if(prices[index]>prices[index-1]){
                profit+=prices[index]-prices[index-1];
            }
        }
        System.out.println(profit);
    }
}
```

SESSION 2:

Problem 1:

Characters in All N String Values

ID:11120 Solved By 717 Users

The program must accept **N** string values as the input. The program must print the common characters in all the N string values in sorted order as the output.
Note: At least one character is always present in all the N string values.

Boundary Condition(s):
2 <= N <= 10^4
1 <= Length of each string <= 10^4

Input Format:
The first line contains N.
The next N lines, each containing a string.

Output Format:
The first line contains the common characters in all the N string values in sorted order.

Example Input/Output 1:
Input:
5
engine
manager
generation
pen
mentor

Output:
en

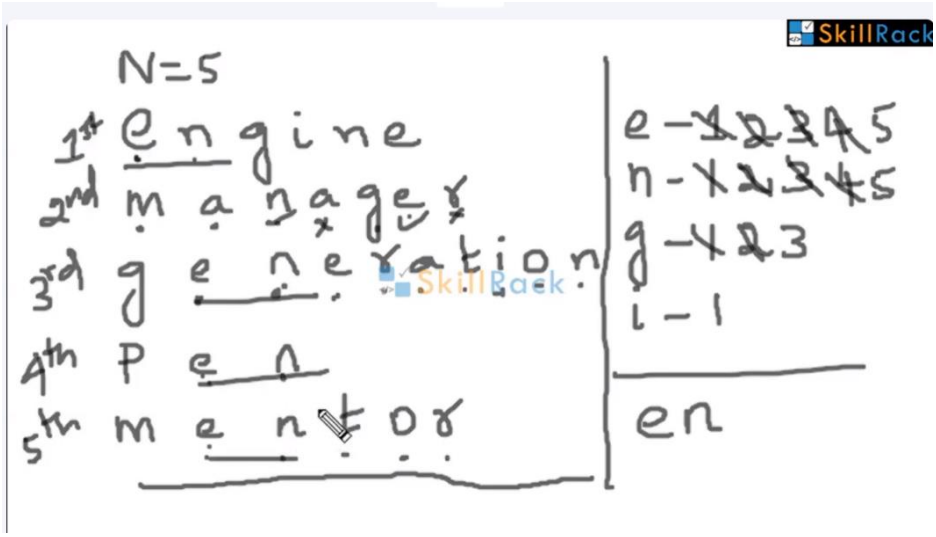
Explanation:
The common characters in all the given **5** string values are **e** and **n**.
Hence the output is en

Example Input/Output 2:
Input:
3
Africa
Australia
Antarctica

Output:
Aair

Example Input/Output 3:
Input:
2
bbBBB
bbBBB

Output:
Bb



Code:

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int N;
    scanf("%d",&N);
    char str[10000];
    int asciiCount[128]={0};
    for(int index=0;index<N;index++){
        scanf("%s",str);
        for(int chindex=0;str[chindex];chindex++){ //loops until the null
character is found in the string
            char ch= str[chindex];
            if(asciiCount[ch]==index){
                asciiCount[ch]++;
            }
        }
    }
    for(int ascii=1;ascii<128;ascii++){
        if(asciiCount[ascii]==N){
            printf("%c",ascii);
        }
    }
}
```

Problem 2:

Characters in at least N-1 String Values

ID:11121 Solved By 687 Users

The program must accept **N** string values as the input. The program must print the common characters that are present in N or N-1 string values in sorted order as the output.
Note: At least one character is always present in N or N-1 string values.

Boundary Condition(s):
3 <= N <= 10^4
1 <= Length of each string <= 10^4

Input Format:
The first line contains N.
The next N lines, each containing a string.

Output Format:
The first line contains the common characters in sorted order that are present in N or N-1 string values in sorted order.

Example Input/Output 1:
Input:
3
orange
apple
pineapple

Output:
aelnp

Explanation:
The common characters that are present in **3** or **2** string values are **a**, **e**, **l**, **n** and **p**.
Hence the output is aelnp

Example Input/Output 2:
Input:
4
HardWork
HomeWork
Hungry
Wood

Output:
HWor

Max Execution Time Limit: 50 millisecs

Code:

```c
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int N;
    scanf("%d",&N);
    char str[10000];
    int asciiCount[128]={0};
    for(int index=0;index<N;index++){
        scanf("%s",str);
```

```
        int currCount[128]={0};
        for(int chindex=0;str[chindex];chindex++){
            char ch=str[chindex];
            if(currCount[ch]==0 && (asciiCount[ch]==index ||
asciiCount[ch]==index-1 )){
                //currCount==0 to avoid duplicate charcters // This can also be
coded as >=index-1 (for ==index || ==index-1)
                currCount[ch]++;
                asciiCount[ch]++;
            }
        }
    }
    for(int ascii=1;ascii<128;ascii++){
        if(asciiCount[ascii]==N || asciiCount[ascii]==N-1){ //this can also be
coded as N>=-1 (for ==N || ==N-1)
            printf("%c",ascii);
        }
    }

}
```

Java Code:

```
import java.util.*;
public class characterInAllStringValues {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int N=sc.nextInt();
        sc.nextLine();

        int[] arr=new int[128];
        for(int i=0;i<N;i++)
        {
            String words=sc.nextLine().trim();
            int []curr=new int[128];
            for(int index=0;index<words.length();index++)
            {
                if(curr[words.charAt(index)]==0 && arr[words.charAt(index)]>=i-1)
                {
                    arr[words.charAt(index)]+=1;
                    curr[words.charAt(index)]++;
                }
```

```
            }
        }
        for(int i=0;i<128;i++)
        {
            if(arr[i]>=N-1)
            {
                System.out.print((char)i);
            }
        }
    }
}
```

SESSION 3:

Problem 1:

Non-Measurable Smallest Weight

ID:11122 Solved By 795 Users

A shop-keeper is having **N** integer values representing the measurement weights. He wishes to find the smallest integer value of weight that cannot be measured using these N weights. Please help the shop keeper by completing the program.

Boundary Condition(s):
1 <= N <= 1000
1 <= Each weight value <= 500

Input Format:
The first line contains N.
The second line contains the N integer values separated by a space.

Output Format:
The first line contains an integer value.

Example Input/Output 1:
Input:
4
2 4 1 10

Output:
8

Explanation:
1, **2**, **4** and **10** can be measured using the given single measurement.
3 - 1 and 2
5 - 1 and 4
6 - 2 and 4
7 - 1, 2 and 4
8 - cannot be measured and hence is printed as the output.

Example Input/Output 2:
Input:
5
1 4 2 4 3

Output:
15

Max Execution Time Limit: 100 millisecs

Code:

```
import java.util.*;
public class nonMeasurableSmallWeight {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in = new Scanner(System.in);
        int N=in.nextInt();
        int weights[] = new int[N];
        for(int index=0;index<N;index++){
            weights[index]=in.nextInt();
        }
        Arrays.sort(weights);
        int measurement=1;
        for(int index=0;index<N;index++){
            if(weights[index]<=measurement){
                measurement+=weights[index];
            }
            else{
                break;
            }
        }
        System.out.println(measurement);
    }
}
```

Problem 2:

Decode Ways - Secret Message

ID:4722 Solved By 769 Users

A top secret message string S containing letters from A-Z (only upper case letters) is encoded to numbers using the following mapping:
'A' -> 1, 'B' -> 2 and so on till Z -> '26'
The program must print the total number of ways in which the received message can be decoded.

Boundary Condition(s):
1 <= Length of S <= 100

Input Format:
The first line contains the string S containing numbers.

Output Format:
The first line contains the number of ways in which S can be decoded.

Example Input/Output 1:
Input:
123

Output:
3

Explanation:
1-A 2-B 3-C 12-L 23-W.
Hence 123 can be decoded as ABC or AW or LC, that is in 3 ways.

Example Input/Output 2:
Input:
1290

Output:
0

Max Execution Time Limit: 500 millisecs

Code:

```
import java.util.*;
public class decodeWays {

    public static void main(String[] args) {
        //Your Code Here
        Scanner in= new Scanner(System.in);
        String str=in.nextLine();
        int ways=1,prevWays=1;
        if(str.charAt(str.length()-1)=='0'){
            ways=0;
        }
        for(int index=str.length()-2;index>=0;index--){
            int backup=prevWays;
            prevWays=ways;
            if(str.charAt(index)=='0'){
                ways=0;
```

```
                continue;
            }
            int twoDigitValue=Integer.parseInt(str.substring(index,index+2));
            if(twoDigitValue<=26){
                ways+=backup;
            }
        }
        System.out.println(ways);
    }
}
```