1::Missing value::

#include<stdio.h>

void main()

{

int i,x=0,y=0,n;

printf("enter the no of numbers to be entered");

scanf("%d",&n);

int a[n-1];

for(i=0;i<n-1;i++)

{

scanf("%d",&a[i]);

}

for(i=0;i<n-1;i++){

x=x^a[i];

}

for(i=0;i<n;i++)

{ y=y^i;

}

printf("missing value is %d",x^y)

}

2::Sum of 2 numbers is K:

#include <stdio.h>

void main()

{

int a[30],i,j,k,sum,n;

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

scanf("%d",&k);

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(a[i]+a[j]==k)

printf("%d %d\n",a[i],a[j]);

}

break;

}

}

3::Array of strings:: to find longest common prefix

#include <stdio.h>

#include<string.h>

void main()

{

char arr[3][10],b[3][10];

char k[30];

int i=0,j=1,d=2,s,n;

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%s",arr[i]);

}

strcpy(b[0],arr[0]);

j=1;

while(j!=n)

{

for(s=0;s<10;s++)

{

if(b[0][s]==arr[j][s])

k[s]=b[0][s];

else

k[s]='\0';

}

strcpy(b[0],arr[j]);

j++;

}

printf("%s",k);

}

4::Minimum characters required to make a string a palindrome::

#include<stdio.h>

#include<string.h>

void main()

{

char arr[30],r[30];

int n,i,j,cnt=0,count=0,end,begin;

scanf("%s",arr);

while (arr[count] != '\0')

count++;

end = count - 1;

for (begin = 0; begin < count; begin++) {

r[begin] = arr[end];

end--;

}

r[begin] = '\0';

if(strcmp(arr,r)==0)

printf("%d",0);

else

{

n=strlen(arr);

j=n-1;

i=0;

while(j!=0)

{

if(arr[i]!=arr[j])

{

arr[j]='\0';

cnt++;

}

if(arr[i]==arr[j])

cnt++;

j--;

}

printf("%d",cnt);

}

}

5::Maximum non negative subarray::

vector<int> Solution::maxset(vector<int> &A) {

int len = A.size();

long long maxSum = 0;

long long curSum = 0;

int startMax = -1;

int endMax = -1;

int start = 0;

int end = 0;

while(end < len) {

if(A[end] >= 0) {

curSum += (long long)A[end];

if(curSum > maxSum) {

maxSum = curSum;

startMax = start;

endMax = end + 1;

} else if(curSum == maxSum) {

if(end + 1 - start > endMax - startMax) {

startMax = start;

endMax = end + 1;

}

}

}else {

start = end + 1;

curSum = 0;

}

end++;

}

vector<int> ans;

ans.clear();

if(startMax == -1 || endMax == -1)

return ans;

for(int i = startMax; i < endMax; ++i)

ans.push\_back(A[i]);

return ans;

}

6::To rearrange array numbers to get largest number::

import java.util.Arrays;

import java.util.Collections;

import java.util.List;

import java.util.\*;

class Util

{

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

int n=s.nextInt();

List<String> numbers=new ArrayList<String>();

for(int i=0;i<n;i++)

{

numbers.add(i,s.next());

}

Collections.sort(numbers, (a, b) -> (b + a).compareTo(a + b));

numbers.stream().forEach(System.out::print);

}

}