Arrays in C++

Assignment Solutions

Q1 - Given an integer array(arr) and its size(n), print the count of odd and even integers present in the array.

For ex: arr[]={1,2,3,4,5} n=5
Output: Odd Numbers = 3
Even Numbers = 2

Explanation: The even numbers present in the array are 2,4 and the odd numbers present in the array are 1,3 and 5.

Code:

```
int count(int arr[],int n)
  {
   //we will count the number of odd elements and subtract it from n which will give
   even elements
   int count_odd=0;
   for(int i=0;i<n;i++)
   {
       if(arr[i]%2!=0)
       {
           count_odd++;
       }
   }
   return count_odd;
 }
```

Q2 - Given an integer array and its size, find the sum of the greatest and the smallest integer present in the array. Here 1< size <101

For ex: arr[]={1,2,3,4,5} n=5
Output: 6

Explanation: The smallest number in the array is 1 and the greatest numbers in the array is 5, so the sum will be 1+5=6

Code:

```
  int sum(int arr[],int n)
{
    int max_val=INT_MIN;
    int min_val=INT_MAX;
    for(int i=0;i<n;i++)
    {
        max_val=max(max_val,arr[i]);
        min_val=min(min_val,arr[i]);
    }
    int ans=max_val+min_val;
    return ans;


}
```

Q3 - Given an integer array and its size, reverse the array and print it. Here 1<size<101

For ex: arr[]={1,2,3,4,5} n=5
Output: 5,4,3,2,1

arr[]={1,1,1,1,1] n=5
Output: 1,1,1,1,1

Code:

```
void rev(int arr[],int n)
{
    int start=0;
    int end=n-1;
    while(start<end)
    {
        int temp=arr[start];
        arr[start]=arr[end];
        arr[end]=temp;
        start++;
        end--;
    }
    for(int i=0;i<n;i++)
    {
        cout<<arr[i]<<" ";
    }
}
```

Q4 - Given two arrays a[] and b[] of same size.Your task is to find the minimum sum of two elements such that they belong to different arrays and are not at the same index. Here 1<size<101

For ex: a[]={5,6,10,4,9}
b[]={1,2,3,4,5}
Output: 5
Explanation: The numbers are 4 from a and 1 from b which makes sum 1+4=5

a[3] + b[0] = 5 is the lowest possible sum amongst all the possible combinations

Algorithm:

Find minimum elements from a[] and b[]. Let these elements be minA and minB respectively.

1. If indexes of minA and minB are not the same, return minA + minB.
2. Else find second minimum elements from two arrays. Let these elements be minA2 and minB2.
   Return min(minA + minB2, minA2 + minB)

Code:

```
int minSum(int a[], int b[], int n)
   {
       int minA=INT_MAX,minB=INT_MAX,indexA=0,indexB=0;
       int min2A=INT_MAX,min2B=INT_MAX;
       for(int i=0;i<n;i++)
       {
           if(a[i]<minA)
           {
               min2A=minA;
               minA=a[i];
               indexA=i;
           }
           else if(a[i]<min2A)
               min2A=a[i];
           if(b[i]<minB)
           {
               min2B=minB;
               minB=b[i];
               indexB=i;
           }
           else if(b[i]<min2B)
               min2B=b[i];
       }
       if(indexA!=indexB)
           return minA+minB;
       return min(min2A+minB,min2B+minA);
  }
```

Q5 - Given an array containing n distinct integers in the range [0,n] (inclusive of both 0 and n) (inclusive of both 0 and n). Find and return the only number of the range that is not present in the array. Here 1<n<101.

Ex: arr=[3,0,1]
Output: 2

n=3, thus the range will be [0,3]

Ex: arr=[8,6,4,2,3,5,0,1]
Output: 7

n=8, thus the range will be [0,8]

Algorithm:

We find the sum of the numbers from [0,n] and subtract the sum of the given array from it and this gives us the missing number in the range.

Code:

```
int val(int arr[],int n)
{
    int missing=-1;
    int sum=0;
    for(int i=0;i<n;i++)
    {
        sum+=arr[i];
    }
    int range_sum=(n)*(n+1)/2;
    missing=range_sum-sum;
    return missing;
}
```

Q6 - Given an integer array containing n elements. Find the element in the array for which all the elements to its left are smaller than it and all the elements to the right of it are larger than it.Here 1<n<101

Ex: arr=[1,6,5,7,10,8,9]
Output: 7

Explanation: Here all the elements to the left of 7 are smaller than it and all the elements to the right of it are greater than it.

Ex: arr=[5,6,2,8,10,9]
Output: -1

Explanation: Here there is no element in the array which satisfies the given condition.

Algorithm:

For each element in our array we find the max element to the left of that element and the minimum element to the right of that element. If our element is greater than the left max and smaller than the right min then we know that this is our answer.

Code:

```
int prefix[N];
        int suffix[N];
        prefix[0]=INT_MIN;
        suffix[N-1]=INT_MAX;
        for(in1;i<N;i++){
            prefix[i]=max(prefix[i-1],arr[i-1]);
        }
        for(int i=N-2;i>=0;i--){
            suffix[i]=min(suffix[i+1],arr[i+1]);
        }
        for(int i=0;i<N;i++){
            if(prefix[i]<arr[i] && arr[i]<suffix[i]){
                return arr[i];
            }t i=
        }
        return -1;
```