**DAY -9**

**PROBLEM 1-** [**493. Reverse Pairs**](https://leetcode.com/problems/reverse-pairs/)

CODE-

class Solution {

public:

int ans = 0;

void merge(vector<int>&a , int low , int mid , int high){

int i = low;

int j = mid + 1;

while(i<=mid && j<=high){

if((long long int)a[i]>(long long int)2\*a[j]){

ans += mid-i+1;

j++;

}

else i++;

}

i = low , j=mid+1;

vector<int>b;

while(i<=mid && j<=high){

if(a[i] < a[j]){

b.push\_back(a[i++]);

}

else b.push\_back(a[j++]);

}

while(i<=mid) b.push\_back(a[i++]);

while(j<=high) b.push\_back(a[j++]);

for(int i=low;i<=high;i++) a[i] = b[i-low];

}

void mergesort(vector<int>&a,int low,int high){

if(low==high) return;

int mid = (low+high)/2;

mergesort(a,low,mid);

mergesort(a,mid+1,high);

merge(a , low , mid , high);

}

int reversePairs(vector<int>& nums) {

mergesort(nums,0,nums.size()-1);

return ans;

}

};

# PROBLEM 2- [20. Valid Parentheses](https://leetcode.com/problems/valid-parentheses/)

# class Solution {

# public:

# bool isValid(string s) {

# stack<char>st;

# // st.push(s[0]);

# for(int i=0;i<s.size();i++)

# {

# if(s[i]=='(' || s[i]=='{' || s[i]=='[')

# st.push(s[i]);

# else

# {

# if(st.empty())return false;

# char c=st.top();

# if((c=='('&&s[i]==')')||(c=='{'&& s[i]=='}')||(c=='['&&s[i]==']'))

# st.pop();

# else

# return false;

# }

# }

# 

# return st.empty();

# }

# };PROBLEM 3- Print all the duplicate characters in a string

**CODE-**

class Solution {

// C++ code to implement the above approach

#include <algorithm>

#include <iostream>

#include <string>

using namespace std;

void printDuplicates(string str)

{

int len = str.length();

// Sorting the string

sort(str.begin(), str.end());

// Loop through the sorted string to find duplicates

for (int i = 0; i < len; i++) {

int count = 1;

// Counting the occurrences of each character

while (i < len - 1 && str[i] == str[i + 1]) {

count++;

i++;

}

// Printing the duplicate character and its count

if (count > 1) {

cout << str[i] << ", count = " << count << endl;

}

}

}

int main()

{

string str = "test string";

printDuplicates(str);

return 0;

}

// This code is contributed by Veerendra\_Singh\_Rajpoot

**PROBLEM 4-**

[**28. Find the Index of the First Occurrence in a String**](https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/)

class Solution {

public:

int strStr(string h, string ni) {

// int n=h.size();

// int n=ni.size();

size\_t found = h.find(ni);

if (found != string::npos)

return found;

return -1;

}

};

**PROBLEM 4-**

**MERGE SORTED ARRAY**

**CODE-**

class Solution {

public:

void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {

vector<int>v;

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

{

v.push\_back(nums1[i]);

}

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

{

v.push\_back(nums2[i]);

}

nums1.clear();

sort(v.begin(),v.end());

for(int i=0;i<v.size();i++)

{

nums1.push\_back(v[i]);

}

}

};