**WEEK V**

1. **Given an unsorted array of alphabets containing duplicate elements. Design an algorithm and implement it using a program to find which alphabet has maximum number of occurrences and print it. (Time Complexity = O(n)) (Hint: Use counting sort)**

**ANSWER I.PROGRAM**

|  |
| --- |
| #include <iostream> |
|  | #include<limits.h> |
|  | using namespace std; |
|  |  |
|  | void count\_sort(char arr[],int n) |
|  | { |
|  | int temp[26]={0}; |
|  | for (int i=0;i<n;i++) |
|  | temp[arr[i]-97]++; |
|  |  |
|  | int maxi=0; |
|  | char res='$'; |
|  | for (int i=0;i<26;i++) |
|  | { |
|  | if (temp[i]>maxi) |
|  | { |
|  | maxi=temp[i]; |
|  | res=i+97; |
|  | } |
|  | } |
|  | if (maxi==1) |
|  | cout<<"No Duplicate Found"<<endl; |
|  | else |
|  | cout<<res<<" - "<<maxi<<endl; |
|  |  |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  |  |
|  | int t; |
|  | cin>>t; |
|  | while (t--) |
|  | { |
|  | int n; |
|  | cin>>n; |
|  | char arr[n]; |
|  | for (int i=0;i<n;i++) |
|  | cin>>arr[i]; |
|  | count\_sort(arr,n); |
|  | } |
|  | return 0; |
|  | } |

**OUTPUT**

**1**

**6**

**a e d s a a**

**a – 3**

1. **Given an unsorted array of integers, design an algorithm and implement it using a program to find whether two elements exist such that their sum is equal to the given key element. (Time Complexity = O(n log n))**

**ANSWER II. PROGRAM**

#include<stdio.h>

#define MAX 100

void find(int A[],int n,int key)

{

int flag=0;

int lb=0 ,ub=n;

while(lb<ub)

{

if(A[lb]+A[ub]==key){

flag=1;

printf("%d\t%d,",A[lb],A[ub]);

lb++;

ub--;

}

else if(A[lb]+A[ub]<key)

lb++;

else

ub--;

}

if(flag==0)

printf("No sequence found");

}

void merge\_sort(int A[],int lb ,int ub)

{

if(lb<ub)

{

int mid=lb+(ub-lb)/2;

merge\_sort(A,lb,mid);

merge\_sort(A,mid+1,ub);

merge(A,lb,mid,ub);

}

}

void merge(int A[],int lb ,int mid ,int ub)

{

int n1=mid+1-lb;

int n2=ub-mid;

int L[MAX],R[MAX];

for(int i=0 ;i<n1 ;i++)

L[i]=A[lb+i];

for(int j=0 ;j<n2 ;j++)

R[j]=A[mid+j+1];

int i=0 ,j=0 ,k=lb;

while(i<n1 && j<n2)

{

if(L[i]<=R[j])

{

A[k]=L[i];

i++;

}

else{

A[k]=R[j];

j++;

}

k++;

}

while(i<n1)

{

A[k]=L[i];

i++;

k++;

}

while(j<n2)

{

A[k]=R[j];

j++;

k++;

}

}

int main()

{

int key;

int t ,n, A[MAX];

scanf("%d",&t);

while(t--)

{

scanf("%d",&n);

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

{

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

}

}

scanf("%d",&key);

merge\_sort(A,0,n-1);

find(A,n,key);

}

**OUTPUT**

**1**

**5**

**1 0 23 56 9**

**24**

**No sequence found**

1. **You have been given two sorted integer arrays of size m and n. Design an algorithm and implement it using a program to find list of elements which are common to both. (Time Complexity = O(m+n))**

**ANSWER III. PROGRAM**

#include<stdio.h>

#define MAX 100

void common(int A1[],int A2[], int m,int n)

{

int i=0 ,j=0 ;

while(i<m && j<n)

{

if(A1[i]<A2[j])

{

i++;

}

else if(A1[i]>A2[j])

{

j++;

}

else

{

printf("%d\t",A2[j++]);

i++;

}

}

}

int main()

{

int t , A1[MAX], A2[MAX], n, m;

scanf("%d",&t);

while(t--)

{

printf("Enter size of A1:");

scanf("%d",&m);

printf("Enter the elements in the array :");

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

{

scanf("%d",&A1[i]);

}

printf("Enter size of A2:");

scanf("%d",&n);

printf("Enter the elements in the array :");

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

{

scanf("%d",&A2[i]);

}

common(A1,A2,m,n);

}

}

**OUTPUT**

**1**

**Enter size of A1:5**

**Enter the elements in the array :10 45 67 89 90**

**Enter size of A2:4**

**Enter the elements in the array :10 45 65 90**

**10 45 90**