**Code:**

#include<iostream>

using namespace std;

//To find largest element of array

int largestElement(int arr[],int n){

int largest=arr[0];

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

if(arr[i]>largest){

largest=arr[i];

}

}

return largest;

}

//To sort elements of array

/\*void sorted(int arr[],int n){

for(int i=0;i<=n-2;i++){

int mini=i;

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

if(arr[j]<arr[mini]){

mini=j;

}

}

int temp=arr[mini];

arr[mini]=arr[i];

arr[i]=temp;

}

}\*/

//to check if is sorted

bool check(int arr[],int n){

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

if(arr[i]<arr[i-1]){

return false;

}

}

return true;

}

//to get second smallest and largest elements

int getSecond(int arr[],int n){

if(n==0 || n==1){

cout<<-1<<" "<<-1<<endl;

}

for(int i=0;i<=n-2;i++){

int mini=i;

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

if(arr[j]<arr[mini]){

mini=j;

}

}

int temp=arr[mini];

arr[mini]=arr[i];

arr[i]=temp;

}

int ssmall=arr[1];

int slargest=arr[n-2];

cout<<slargest<<" "<<ssmall<<endl;

return 0;

}

int main(){

int n=6;

int arr[n]={2,5,8,9,13,12};

bool ans =check(arr,n);

if(ans){

cout<<"True"<<endl;

}else{

cout<<"False"<<endl;

}

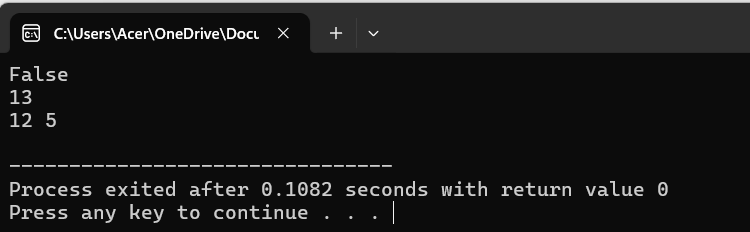
int res=largestElement(arr,n);

cout<<res<<endl;

getSecond(arr,n);

return 0;

}



#include<iostream>

using namespace std;

//to remove duplicates

int removeDuplicates(int arr[],int n){

int i=0;

for(int j=1;j<n;j++){

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

arr[i+1]=arr[j];

i++;

}

}

return i+1;

}

//to rotate left

int rotate(int arr[],int n){

int temp=arr[0];

for(int i=1;i<n;i++){

arr[i-1]=arr[i];

}

arr[n-1]=temp;

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

cout<<arr[i]<<" ";

}

}

//to searche an element

int linearSearch(int arr[],int n,int k){

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

if(arr[i]==k){

return 1;

}

}

return -1;

}

//to sort array of 0's,1's and 2's

void sortColors(int arr1[],int s){

int c1=0,c2=0,c3=0;

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

if(arr1[i]==0){

c1++;

}

}

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

if(arr1[i]==1){

c2++;

}

}

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

if(arr1[i]==2){

c3++;

}

}

for(int i=0;i<c1;i++){

arr1[i]=0;

}

for(int i=c1;i<c1+c2;i++){

arr1[i]=1;

}

for(int i=c1+c2;i<s;i++){

arr1[i]=2;

}

}

//to find sum of maximum subarray

int maxsubarray(int arr1[],int s){

long long maxi=LONG\_MIN;

long long sum=0;

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

sum+=arr1[i];

if(sum>maxi){

maxi=sum;

}

if(sum<0){

sum=0;

}

}

return maxi;

}

//to rearrange an array

int rearrange(int arr2[],int s){

int ar[s];

int pos=0,neg=1;

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

if(arr2[i]<0){

ar[neg]=arr2[i];

neg+=2;

}else{

ar[pos]=arr2[i];

pos+=2;

}

}

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

cout<<ar[i]<<" ";

}

}

int main(){

int s=8;

int arr1[s]={2,1,2,0,0,1,2,2};

int arr2[s]={2,4,-1,5,-5,9,-3,-1};

int arr[]={1,1,2,4,4,6};

int k=8;

int n=sizeof(arr)/sizeof(arr[0]);

int res=removeDuplicates(arr,n);

cout<<"After remove duplicates:\n";

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

cout<<arr[i]<<" ";

}

cout<<endl;

cout<<"After rotate:\n";

rotate(arr,n);

cout<<endl;

cout<<"Linear search:\n";

int ans=linearSearch(arr,n,k);

cout<<ans<<endl;

cout<<"Array sorted:\n";

sortColors(arr1,s);

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

cout<<arr1[i]<<" ";

}

cout<<endl;

cout<<"Maximum of subarray:\n";

int as=maxsubarray(arr1,s);

cout<<as<<endl;

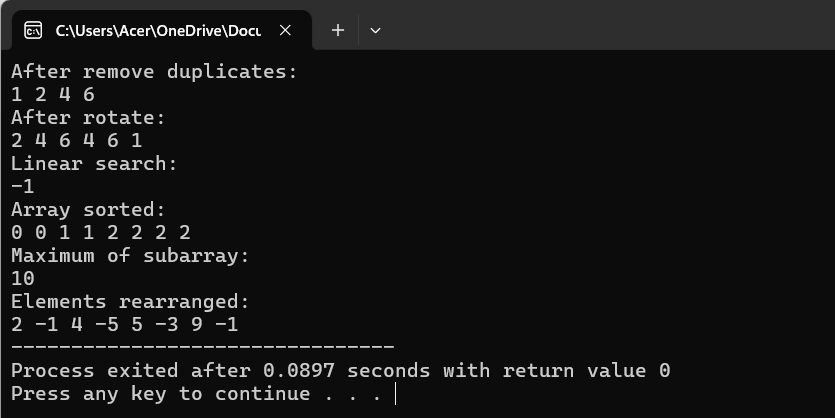
cout<<"Elements rearranged:\n";

rearrange(arr2,s);

return 0;

}

**Output:**



**Code:**

#include<iostream>

#include<bits/stdc++.h>

using namespace std;

//Count inversions

int numberOfInversions(int arr[],int n){

int count=0;

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

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

if(arr[i]>arr[j]){

count++;

}

}

}

return count;

}

//Reverse pairs

int team(int arr[],int n){

int count=0;

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

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

if(arr[i]>2\*arr[j]){

count++;

}

}

}

return count;

}

//Maximum product subarray

int subarray(int arr[],int n){

int result=arr[0];

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

int p=arr[i];

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

result=max(result,p);

p\*=arr[j];

}

result=max(result,p);

}

return result;

}

//Finding missing and repeating numbers

int findmissingrepeating(int arr[],int n){

int repeat=-1,missing=-1;

for(int i=1;i<=n;i++){

int count=0;

for(int j=0;j<n;j++){

if(arr[j]==i){

count++;

}

}

if(count==2){

repeat=i;

}else if(count==0){

missing=i;

}

if(repeat != -1 && missing != -1){

break;

}

}

return(repeat,missing);

}

//Count subarrays with given sum

int subarraysum(int arr[],int n,int k){

int count=0;

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

int sum=0;

for(int j=i;j<n;j++){

sum+=arr[j];

if(sum==k){

count++;

}

}

}

return count;

}

int main(){

int n=6;

int arr[n]={2,4,5,7,8,1};

int k=5;

int res=numberOfInversions(arr,n);

cout<<"The number of inversions:\n";

cout<<res<<endl;

cout<<"The number of reverse pairs:\n";

int r=team(arr,n);

cout<<r<<endl;

int re=subarray(arr,n);

cout<<"Maximum product of subarray:\n";

cout<<re<<endl;

int er=findmissingrepeating(arr,n);

cout<<"Missing and repeating elements:\n";

cout<<er<<endl;

int ser=subarraysum(arr,n,k);

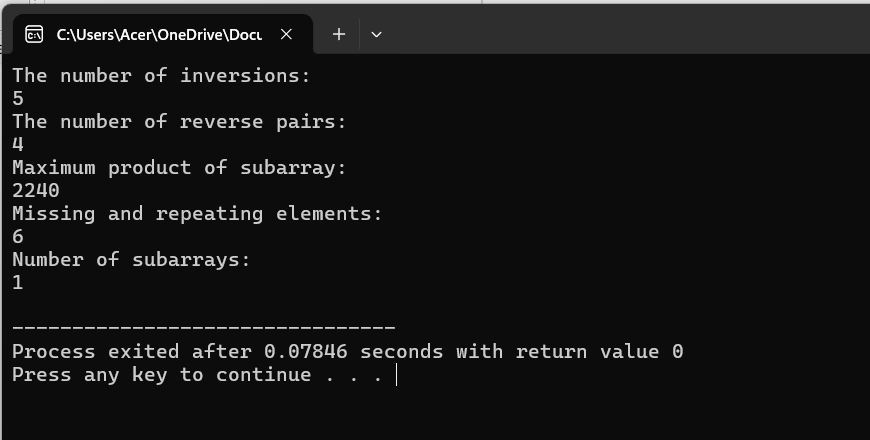
cout<<"Number of subarrays:\n";

cout<<ser<<endl;

return 0;

}

**Output:**



**Code:**

#include<iostream>

using namespace std;

//Union of two sorted arrays

void unionn(int arr1[],int arr2[],int arr[],int n,int m){

int i=0,j=0;

while(i<n && j<m){

if(arr1[i]<arr2[j]){

arr[i+j]=arr1[i];

i++;

}

else if(arr2[j]<arr1[i]){

arr[i+j]=arr2[j];

j++;

}else{

arr[i+j]=arr2[j];

j++;

i++;

}

}

while(i < n){

arr[i+j]=arr1[i];

i++;

}

while(j < m){

arr[i+j]=arr2[j];

j++;

}

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

cout<<arr[i]<<" ";

}

}

//Count maximum subarray of ones

int findmax(int arr3[],int k){

int count=0;

int maxi=0;

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

if(arr3[i]==1){

count++;

}else{

count=0;

}

if(count>maxi){

maxi=count;

}

}

return maxi;

}

int main(){

int n=4,m=5;

int arr1[n]={2,4,5,7};

int arr2[m]={1,3,6,8,10};

int arr[n+m];

cout<<"Union of two sorted arrays:\n";

unionn(arr1,arr2,arr,n,m);

cout<<"\n";

cout<<"Count maximum subarray of ones:\n";

int k=6;

int arr3[k]={1,1,0,1,1,1};

int res=findmax(arr3,k);

cout<<res<<endl;

return 0;

}

**Output:**

