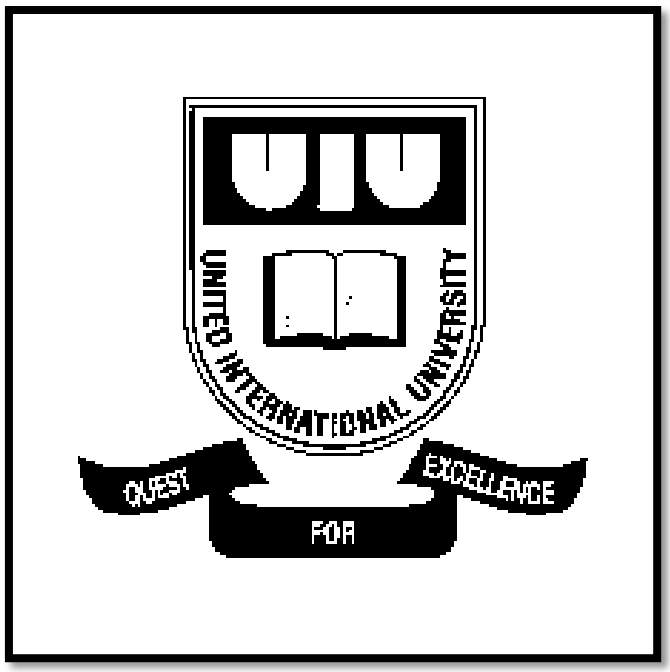


Assignment:01

Data Structure and Algorithms I Laboratory

Course Code: CSE 218

Date: 23/07/2021

**Submitted To**

G. M. Monjur Morshed
Senior Software Analyst

**Submitted By**

Sunny Sutradhar
ID: 011201084
Section: D

**1. Write a program to find duplicate number(s) on Integer array.**

**Solution:**

```
#include<bits/stdc++.h>
using namespace std;
void duplicate_finder(int a[],int n){
    int max=0;
    for(int i=0;i<n;i++){
        if(a[i]>max)max=a[i];
    }
    int duplicate[max+1]={0};
    for(int i=0;i<n;i++){
        duplicate[a[i]]++;
    }
    for(int i=0;i<=max;i++){
        if(duplicate[i]>1)cout<<i<<" ";
    }
    cout<<endl;
}
int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
       cin>>a[i];
    }
    duplicate_finder(a,n);
}
```

**2. WAP that will take n integers into an array A. Now remove all duplicates numbers from that array. Finally print all elements from that array.**

**Solution:**

```
#include<bits/stdc++.h>
using namespace std;

int duplicate_remover(int a[],int n){
    int index=0,max=0;
    for(int i=0;i<n;i++){
        if(a[i]>max)max=a[i];
    }
    int duplicate[max+1]={0};
    for(int i=0;i<n;i++){
        duplicate[a[i]]++;
    }
    for(int i=0;i<=max;i++){
         if(duplicate[i]>0)a[index++]=i;
    }
    return index;
}
int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
       cin>>a[i];
    }
    int index =duplicate_remover(a,n);
    for(int i=0;i<index;i++){
      cout<<a[i]<<" ";
    }
    cout<<endl;
}
```

**3. Write a program to find intersection of two sorted array. For example, if the two sorted arrays as input are {21, 34, 41, 22, 35} and {61, 34, 45, 21, 11}, it should return an intersection array with numbers {34, 21}. For the sake of this Problem, you can assume that numbers in each integer array are unique.**

**Solution:**

```
#include<bits/stdc++.h>
using namespace std;

int intersection(int a[],int n,int b[],int m,int intersectionArray[]){
    int index=0;
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(a[i]==b[j]){
               intersectionArray[index++]=a[i];
            }
        }
    }  return index;
}
int main(){
    int n,m;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
       cin>>a[i];
    }
    cin>>m;
    int b[m],intersectionArray[m];
    for(int i=0;i<m;i++){
      cin>>b[i];
    }
    int index = intersection(a,n,b,m,intersectionArray);
    for(int i=0;i<index;i++){
      cout<<intersectionArray[i]<<" ";
    }cout<<endl;
```

```
}
```

**4. Write a program to find kth smallest element in unsorted array? You are given an unsorted array of numbers and k, you need to find the kth smallest number in the array. For example if given array is {1, 2, 3, 9, 4} and k=2 then you need to find the 2nd smallest number in the array, which is 2.**

**Solution:**

```
#include<bits/stdc++.h>
using namespace std;

int kThSmallestElement(int a[],int n,int k){
    for(int i=0;i<n;i++){
       int item = a[i];
       int j = i-1;
       while(j>=0 && a[j]>item){
           a[j+1] = a[j];
           j --;
       }
       a[j+1]=item;
    }
    return a[k-1];
}
int main(){
    int n,k;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
       cin>>a[i];
    }
    cin>>k;
    cout<<kThSmallestElement(a,n,k);
}
```

**5. Write a program to reverse array in place.**

**Solution:**

```
#include<bits/stdc++.h>
```

```
using namespace std;
void reverseArray(int a[],int n){
    int j=n-1;
    for(int i=0;i<n/2;i++){
        int temp = a[i];
            a[i]=a[j];
            a[j--]=temp;
      }
}
int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
       cin>>a[i];
    }
    reverseArray(a,n);
    for(int i=0;i<n;i++){
      cout<<a[i]<<" ";
    }cout<<endl;
}
```

**6. Write a program that will randomly generate 500 integer numbers in the range -249 to 250. Write those numbers in a text file named "in.txt". Now Sort (in ascending order) all the integer numbers in the file "in.txt" using Bubble Sort. Save the sorted output into another text file named "out.txt". (Try to write separate function for SWAP and BUBBLE_SORT and call them from main function.)**

**Solution:**

```
#include<bits/stdc++.h>
using namespace std;
void swap(int *a,int *b){
    int temp = *a;
       *a = *b;
       *b = temp;
```

```
}
void bubbleSort(int a[],int n){
    for(int i=0;i<n-1;i++){
      for(int j=0;j<n-i-1;j++){
            if(a[j]>a[j+1])swap(&a[j],&a[j+1]);
      }
    }
}
int main(){
   int a[500];
   FILE *in = fopen("in.txt","w");
   if(!in){
      cout<<"FILE NOT FOUND!"<<endl;
   }
   else{
       srand(time(0));
      for(int i=0;i<500;i++){
         a[i] = -249 + rand()%250;
         fprintf(in, "%d ",a[i] );
      }
      fprintf(in,"\n");
   }
   fclose(in);
   bubbleSort(a,500);
   FILE *out = fopen("out.txt","w");
   if(!out){
      cout<<"FILE NOT FOUND!"<<endl;
   }
   else{
      for(int i=0;i<500;i++){
         fprintf(out, "%d ",a[i] );
      }
      fprintf(out,"\n");
   }
```

```
    fclose(out);
}
```

**7. Write a program that will randomly generate 1000 uppercase letters. Write those alphabets in a text file named “in.txt”. Now Sort (in ascending order) all the alphabets in the file “in.txt” using insertion sort. Use separate function for REPLACEMENT_SORT. Show the sorted output in your console window.**

**Solution:**

```
#include<bits/stdc++.h>
using namespace std;
void replaceMentSort(char c[],int n){
      for(int i=0;i<n-1;i++){
         for(int j=i+1;j<n;j++){
            if(c[i]>c[j]){
            char swap =c[i];
                c[i]=c[j];
                c[j]=swap;
          }
         }
      }
}
void insertionSortWithTXT(){
   char c[1000];
   FILE *in;
    in = fopen("in.txt","r");
    if(!in){
      cout<<"FILE DOESN,T EXIST!"<<endl;
    }else{
      for(int i=0;i<1000;i++){
      fscanf(in, "%c",&c[i]);
     }
    }
    fclose(in);
    for(int i=0;i<1000;i++){
```

```
        char key = c[i];
        int j = i-1;
        while(j>=0 && c[j]>key){
            c[j+1] = c[j];
            j--;
        }
        c[j+1]=key;
    }
    in = fopen("in.txt","w");
    if(!in){
      cout<<"FILE DOESN,T EXIST!"<<endl;
    }
    else{
      for(int i=0;i<1000;i++){
      fprintf(in, "%c ", c[i]);
     }
    }
    fclose(in);
}
 int main(){
    FILE *in;
    char c[1000];
    in = fopen("in.txt","w");
    if(!in){
      cout<<"FILE DOESN,T EXIST!"<<endl;
    }
    else{
      srand(time(0));
      for(int i=0;i<1000;i++){
      c[i] = 'A' + (rand()%26);
      fprintf(in, "%c", c[i]);
     }
    }
    fclose(in);
```

```
    insertionSortWithTXT();
    replaceMentSort(c,1000);
    for(int i=0;i<1000;i++){
      cout<<c[i]<<endl;
    }
```

**8. WAP that will take n positive integers into an array A. Now find all the integers that are divisible by 3 and replace them by -1 in array A. Finally show all elements of array A.**

**Solution:**

```
#include<bits/stdc++.h>
using namespace std;
void divisibleBy3(int a[],int n){
    for(int i=0;i<n;i++){
        if(a[i]%3==0)a[i]=-1;
    }
}
int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
      cin>>a[i];
    }
    divisibleBy3(a,n);
    for(int i=0;i<n;i++){
      cout<<a[i]<<endl;
    }
}
```

**9. You are given some coins. You have to take some coins from the given coins such that the sum of the coins you took have a value strictly larger than the sum of the rest of the coins. However, you have to take minimum number of coins while satisfying this condition.**

**Solution:**

```
#include<bits/stdc++.h>
```

```
using namespace std;

int minCoin(int a[],int n){
    int totalSum=0,maxSum=0;
    for(int i=0;i<n;i++){
        totalSum+=a[i];
    }
    for(int i=0;i<n;i++){
        for(int j=i+1;j<n-1;j++){
            if(a[i]<a[j]){
                swap(a[i],a[j]);
            }
        }
    }
    for(int i=0;i<n;i++){
        maxSum+=a[i];
        totalSum-=a[i];
        if(maxSum>totalSum)return i+1;
    }
}
int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
      cin>>a[i];
    }
    int ans = minCoin(a,n);
    cout<<ans<<endl;
}
```

**10. Given a list of unsorted integers, find the pair of elements that have the smallest absolute difference between them. If there are multiple pairs, find them all.**

**Solution:**

```
#include<bits/stdc++.h>
```

```
using namespace std;

void smallestPair(int a[],int n){
    int value=abs(a[0]-a[1]);
    int p1=a[0],p2=a[1];
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
           if(abs(a[i]-a[j])<=value){
              p1 = a[i];
              p2 = a[j];
              value = abs(a[i]-a[j]);
           }
        }
    }
    cout<<p1<<" "<<p2<<endl;
}
int main(){
    int n;
    cin>>n;
    int a[n];
    for(int i=0;i<n;i++){
      cin>>a[i];
    }
    smallestPair(a,n);

}
```