/\* WAP in C++ to perform basic operations on **1-D arrays** .\*/

#include<iostream.h>

#include<limits.h>

#include<process.h>

#include<conio.h>

/\*--------------------------------------------------------------Function declaration -------------------------------------------------------\*/

int linear\_search(int A[ ],int n, int p);

void binary\_search(int A[ ],int n, int p);

void selection\_sort(int A[ ],int n);

void bubble\_sort(int A[ ],int n);

void insertion\_sort(int A[ ], int n);

void merge(int low,int mid,int high);

void merge\_sort(int low,int high);

void merging(int A[],int B[], int n, int m, char c);

void insert\_element(int A[], int n);

void delete\_element(int A[], int n);

int A[50],n,mg=0; // global variable

/\*--------------------------------------------------------------- Main's body ----------------------------------------------------------------------\*/

void main()

{clrscr();

int ch,o=0,h=0,t=0,k,m;

int B[50],p,subscript;

char choice;

cout<<"Enter the Size of array : ";

cin>>n;

cout<<"\n\nEnter the elements : \n\n";

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

cin>>A[i];

cout<<"\n\nArray formed is : ";

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

cout<<A[i]<<" ";

do{

lm:

cout<<"\n\n\n\nChoose from the following : ";

cout<<"\n\n\n1. Search element in an array ";

cout<<"\n\n2. Sort the array ";

cout<<"\n\n3. Merge two arrays ";

cout<<"\n\n4. Insert element in array ";

cout<<"\n\n5. Delete element in array ";

cout<<"\n\n6. Exit ";

cout<<"\n\n\nEnter your choice : ";

cin>>ch;

switch(ch)

{

case 1 : l:

cout<<"\n\n\t1. Linear search ";

cout<<"\n\n\t2. Binary search ";

cout<<"\n\n\tEnter your choice : ";

cin>>ch;

switch(ch)

{

case 1 : cout<<"\n\nEnter element to be searched : ";

cin>>p;

subscript=linear\_search(A,n,p);

if(subscript== -1)

cout<<"\n\nRequested element not found.";

else

cout<<"\n\nSearch Successful.";

cout<<"\n\n\nThe requested element is "<<p<<". \n\nSubscript = "<<subscript<<"\nPosition in array = "<<subscript+1;

break;

case 2 : cout<<"\n\nEnter element to be searched : ";

cin>>p;

binary\_search(A,n,p); break;

default: cout<<"\n\nPlease enter desired keyword.";

goto l;

}

break;

case 2 : k:

cout<<"\n\n\t1. Selection Sort ";

cout<<"\n\n\t2. Bubble Sort ";

cout<<"\n\n\t3. Insertion sort ";

cout<<"\n\n\t4. Merge Sort ";

cout<<"\n\n\tEnter your choice : ";

cin>>ch;

switch(ch)

{

case 1 : if(o==1)

cout<<"\n\nArray already sorted using Selection sort ";

else

{if(h==0&&t==0)

{

selection\_sort(A,n);

cout<<"\n\n\n\nSorted array is : \n\n";

for(k=0; k<n; k++)

cout<<A[k]<<" ";

o=1;

}

else if(h==1&&t==0)

cout<<"\n\nArray already sorted using Bubble sort ";

else

cout<<"\n\nArray already sorted using Insertion sort ";

}

break;

case 2 : if(h==1)

cout<<"\n\nArray already sorted using Insertion sort ";

else

{if(o==0&&t==0)

{ bubble\_sort(A,n);

cout<<"\n\n\n\nSorted array is : \n\n";

for( k=0; k<n; k++)

cout<<A[k]<<" ";

h=1;

}

else if(t==1&&o==0)

cout<<"\n\nArray already sorted using Insertion sort ";

else

cout<<"\n\nArray already sorted using Selection sort ";

}

break;

case 3: if(t==1)

cout<<"\n\nArray already sorted using Insertion sort ";

else

{

if(h==0&&o==0)

{

insertion\_sort(A,n);

t=1;

}

else if(h==1&&o==0)

cout<<"\n\nArray already sorted using Bubble sort ";

else

cout<<"\n\nArray already sorted using selection sort ";

}

break;

case 4 : merge\_sort(0,n);

cout<<"\n\nArray after merge sort : \n\n";

for(int z=1; z<=n; z++)

cout<<A[z]<<" ";

break;

default: cout<<"\n\nPlease enter desired keyword.";

goto k;

}

break;

case 3: cout<<"\n\nEnter size of second array : ";

cin>>m;

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

cin>>B[i];

cout<<"\n\nFirst array : \n\n";

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

cout<<A[i]<<" ";

cout<<"\n\nSecond array : \n\n";

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

cout<<B[i]<<" ";

m:

cout<<"\n\n\t1. Merge in ascending order ";

cout<<"\n\n\t2. Merge in descending order ";

cout<<"\n\n\tEnter your choice : ";

cin>>ch; mg=1;

switch(ch)

{

case 1: merging(A,B,n,m,'a'); break;

case 2: merging(A,B,n,m,'d'); break;

default: cout<<"\n\nEnter Valid keyword ";

goto m;

}

break;

case 4: insert\_element(A,n);

break;

case 5: delete\_element(A,n);break;

case 6: exit(0); break;

default : cout<<"\n\nPlease enter desired keyword : ";

goto lm;

}

cout<<"\n\nWant to choose from menu again : ";

cin>>choice;

}while(choice=='y'||choice=='Y');

getch();

} // end of main

/\*----------------------------------------------------------- Function Definitions --------------------------------------------------------------\*/

int linear\_search(int A[], int n, int p) //function definition

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

{if (A[i]==p)

return i;

}

return -1;

}

void binary\_search(int A[], int n, int p)

{ int L,U,mid; char ch;

lb: L=0; U=n-1;

while(L<=U) //i.e loop will continue if L<=u. if L>U loop will end

{ mid=(L+U)/2;

if(A[mid]==p)

{ cout<<"\n\nElement "<<p<<" found. Search Successful.";

cout<<"\n\nSubscript = "<<mid<<" \n\nPosition = "<<mid+1;

break;

}

else if(p<=A[mid])

U=mid-1;

else

L=mid+1;

}//end of while loop

if(L>U)

{cout<<"\n\nUnsuccessful search.";

cout<<"\n\n\n\nWant to search again. : "; cin>>ch;

if(ch=='y'||ch=='Y')

{cout<<"\n\n\n\nEnter the element to be searched : ";

cin>>p;

goto lb;}

else

exit(1);

}

}

void selection\_sort(int A[], int n)

{int small; int k,count=0;

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

{ small=A[i]; count++;

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

{

if(A[j]<small)

{small=A[j];

A[j]=A[i];

A[i]=small;

}

}

if(mg==0)

{

cout<<"\n\nArray after iteration "<<count<<" is :\n\n";

for(k=0; k<n; k++)

cout<<A[k]<<" ";

}

}

}

void bubble\_sort (int A[], int n)

{ int temp; int count=0;

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

{

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

{ if(A[j+1]<A[j])

{ count++;

temp=A[j+1];

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

A[j]=temp;

cout<<"\n\nArray for iteration "<<count<<" is : \n\n";

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

cout<<A[k]<<" ";

}

}

}

}

void insertion\_sort(int A[], int n)

{int j,t;

A[n]=0;

for(int i=n; i>=0; i--) //shifting each element to its succesive position

A[i]=A[i-1];

A[0]=INT\_MIN; // INT\_MIN= -32768

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

{ t=A[i];

j=i-1;

while(t<A[j])

{

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

j--;

}

A[j+1]=t;

cout<<"\n\nArray after iteration "<<i<<" => \n\n";

for(int k=1; k<=n; k++)

cout<<A[k]<<" ";

cout<<"\n\n";

}

cout<<"\n\n\nSorted array is : \n\n";

for(int k=1; k<=n; k++)

cout<<A[k]<<" ";

}

void merge\_sort(int low,int high)

{

int mid;

if(low<high)

{

mid=(low+high)/2;

merge\_sort(low,mid);

merge\_sort(mid+1,high);

merge(low,mid,high);

}

}

void merge(int low,int mid,int high)

{

int h,i,j,b[50],k;

h=low;

i=low;

j=mid+1;

while((h<=mid)&&(j<=high))

{

if(A[h]<=A[j])

{

b[i]=A[h];

h++;

}

else

{

b[i]=A[j];

j++;

}

i++;

}

if(h>mid)

{

for(k=j;k<=high;k++)

{

b[i]=A[k];

i++;

}

}

else

{

for(k=h;k<=mid;k++)

{

b[i]=A[k];

i++;

}

}

for(k=low;k<=high;k++) A[k]=b[k];

}

void merging(int A[],int B[],int n,int m, char cha)

{mg=1;

int c[80];

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

{

c[i]=A[i];

}

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

{

c[n+i]=B[i];

}

int p=m+n;

switch(cha)

{ case 'a' : cout<<"\n\nArray formed after merging in ascending order : \n\n";

selection\_sort(c,p);

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

{

cout<<c[i]<<" ";

}

break;

case 'd' : cout<<"\n\nArray formed after merging in descending order : \n\n";

selection\_sort(c,p);

for(i=p-1; i>=0; i--)

{

cout<<c[i]<<" ";

}

break;

}

}

void insert\_element(int A[], int n)

{ int p,pos;char choice;

do

{if(n>=50||n<1)

cout<<"\n\nArray Overflow.";

else

cout<<"\n\nEnter element along with its position : \n\n";

cout<<"Element = "; cin>>p;

cout<<"\n\nPosition = "; cin>>pos;

for(int i=n; i>=pos; i--)

{

A[i]=A[i-1];

}

n++;

A[n]=0;

A[pos-1]=p;

cout<<"\n\nArray formed : ";

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

cout<<A[i]<<" ";

cout<<"\n\nWant to insert again : "; cin>>choice;

}while(choice=='y'||choice=='Y');

}

void delete\_element(int A[], int n)

{ int pos,flag=1;

char choice;

do{

l:

cout<<"\n\n\nEnter position of element to delete it : ";

cin>>pos;

if(pos-1>n-1)

{

cout<<"\n\nInvalid ";

goto l;

}

else

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

{

if(pos-1 ==i)

{flag=0;

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

{

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

}

n--;

}

}

if(flag)

{

cout<<"\n\nInvalid ... Enter again : ";

goto l;

}

cout<<"\n\n\nArray formed : ";

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

cout<<A[i]<<" ";

cout<<"\n\n\nWant to delete again : ";

cin>>choice;

}while(choice=='y'||choice=='Y');

}

**Output:**

Enter the Size of array : 5

Enter the elements :

1

3

2

4

0

Array formed is : 1 3 2 4 0

Choose from the following :

1. Search element in an array

2. Sort the array

3. Merge two arrays

4. Insert element in array

5. Delete element in array

6. Exit

Enter your choice : 1

1. Linear search

2. Binary search

Enter your choice : 1

Enter element to be searched :4

Search Successful.

The requested element is 4.

Subscript = 3

Position in array = 4

Want to choose from menu again :Y

Choose from the following :

1. Search element in an array

2. Sort the array

3. Merge two arrays

4. Insert element in array

5. Delete element in array

6. Exit

Enter your choice : 2

1. Selection Sort

2. Bubble Sort

3. Insertion sort

4. Merge Sort

Enter your choice : 2

Array for iteration 1 is :

1 2 3 4 0

Array for iteration 2 is :

1 2 3 0 4

Array for iteration 3 is :

1 2 0 3 4

Array for iteration 4 is :

1 0 2 3 4

Array for iteration 5 is :

0 1 2 3 4

Sorted array is :

0 1 2 3 4

Want to choose from menu again : Y

Choose from the following :

1. Search element in an array

2. Sort the array

3. Merge two arrays

4. Insert element in array

5. Delete element in array

6. Exit

Enter your choice : 2

1. Selection Sort

2. Bubble Sort

3. Insertion sort

4. Merge Sort

Enter your choice : 4

Array after merge sort :

0 1 2 3 4

Want to choose from menu again :Y

Choose from the following :

1. Search element in an array

2. Sort the array

3. Merge two arrays

4. Insert element in array

5. Delete element in array

6. Exit

Enter your choice : 3

Enter your choice : 3

Enter size of second array : 5

1

0

8

9

2

First array :

1 3 2 4 0

Second array :

1 0 8 9 2

1. Merge in ascending order

2. Merge in descending order

Enter your choice : 1

Array formed after merging in ascending order :

0 0 1 1 2 2 3 4 8 9

Want to choose from menu again :Y

1. Seaarch element in an array

2. Sort the array

3. Merge two arrays

4. Insert element in array

5. Delete element in array

6. Exit

Enter your choice : 4

Enter element along with its position :

Element = 6

Position = 2

Array formed : 1 6 3 2 4 0

Want to insert again : n

Want to choose from menu again : y

Choose from the following :

1. Search element in an array

2. Sort the array

3. Merge two arrays

4. Insert element in array

5. Delete element in array

6. Exit

Enter your choice : 5

Enter position of element to delete it : 6

Invalid

Enter position of element to delete it : 5

Array formed : 1 6 3 2

Want to delete again : N

Want to choose from menu again : y

Choose from the following :

1. Search element in an array

2. Sort the array

3. Merge two arrays

4. Insert element in array

5. Delete element in array

6. Exit

Enter your choice : 6