1.Bubble Sort

#include<iostream>

using namespace std;

int main ()

{

int i, j,temp,pass=0;

int a[8] = {9, 20, 6, 10, 14, 8, 60, 11};

cout <<"Input list ...\n";

for(i = 0; i<8; i++) {

cout <<a[i]<<"\t";

}

cout<<endl;

for(i = 0; i<8; i++) {

for(j = i+1; j<8; j++)

{

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

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

pass++;

}

cout <<"Sorted Element List ...\n";

for(i = 0; i<8;i++) {

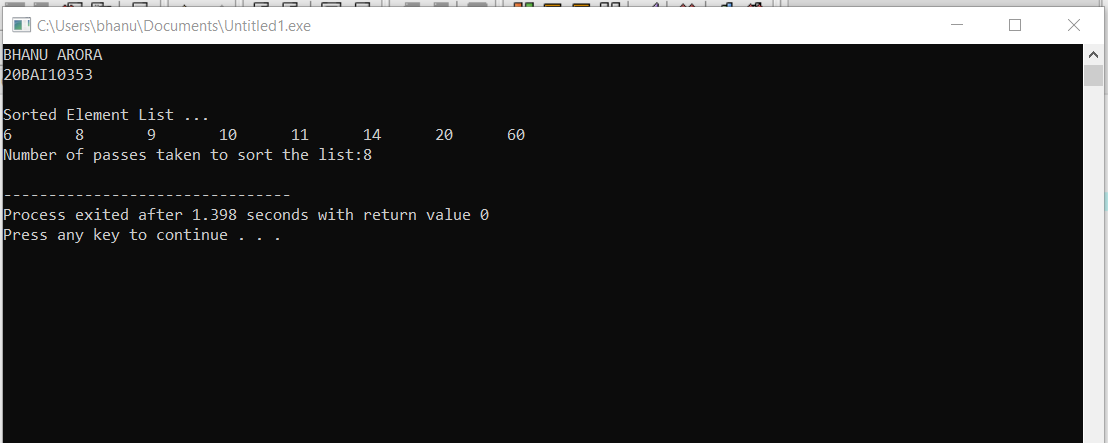
cout <<a[i]<<"\t";

}

cout<<"\nNumber of passes taken to sort the list:"<<pass<<endl;

return 0;

}



2.Selection Sort

#include<iostream>

using namespace std;

int main ()

{

cout<<"BHANU ARORA"<<endl;

cout<<"20BAI10353"<<endl;

int i, j,temp,pass=0;

int a[8] = {9, 20, 6, 10, 14, 8, 60, 11};

cout<<endl;

for(i = 0; i<8; i++) {

for(j = i+1; j<8; j++)

{

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

temp = a[j];

a[j] = a[i];

a[i] = temp;

}

}

pass++;

}

cout <<"Sorted Element List ...\n";

for(i = 0; i<8;i++) {

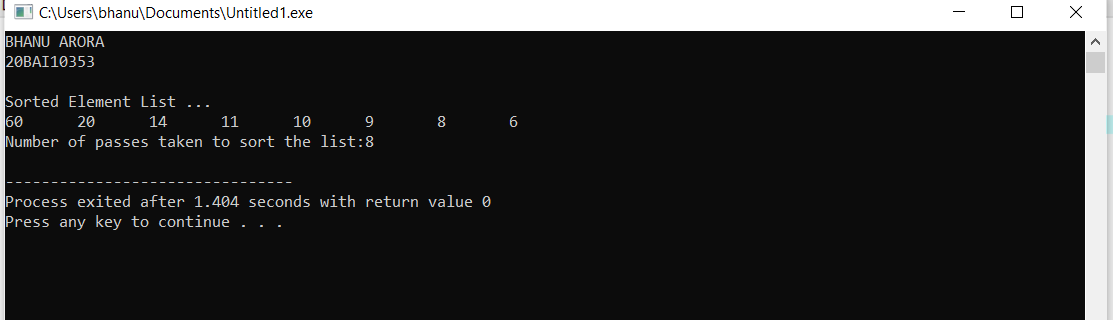
cout <<a[i]<<"\t";

}

cout<<"\nNumber of passes taken to sort the list:"<<pass<<endl;

return 0;

}



3.Insertion

using namespace std;

int main()

{

cout<<"BHANU ARORA"<<endl;

cout<<"20BAI10353"<<endl;

int i, j, k, temp;

int st[8] = {9, 20, 6, 10, 14, 8, 60, 11} ;

for (i = 1; i < 8; i++)

{

for (j = i; j >= 1; j--)

{

if (st[j] < st[j-1])

{

temp = st[j];

st[j] = st[j-1];

st[j-1] = temp;

}

else

break;

}

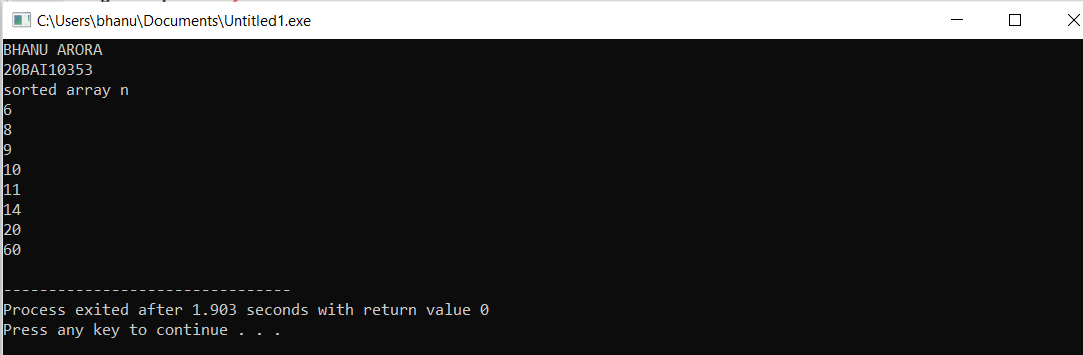
}

cout<<"sorted array n "<<endl;

for (k = 0; k < 8 ; k++)

cout<<st[k]<<endl;

}



4.Merge

#include<iostream>

using namespace std;

void swapping(int &a, int &b) {

int temp;

temp = a;

a = b;

b = temp;

}

void display(int \*array, int size) {

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

cout << array[i] << " ";

cout << endl;

}

void merge(int \*array, int l, int m, int r) {

int i, j, k, nl, nr;

nl = m-l+1; nr = r-m;

int larr[nl], rarr[nr];

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

larr[i] = array[l+i];

for(j = 0; j<nr; j++)

rarr[j] = array[m+1+j];

i = 0; j = 0; k = l;

while(i < nl && j<nr) {

if(larr[i] <= rarr[j]) {

array[k] = larr[i];

i++;

}else{

array[k] = rarr[j];

j++;

}

k++;

}

while(i<nl) {

array[k] = larr[i];

i++; k++;

}

while(j<nr) {

array[k] = rarr[j];

j++; k++;

}

}

void mergeSort(int \*array, int l, int r) {

int m;

if(l < r) {

int m = l+(r-l)/2;

mergeSort(array, l, m);

mergeSort(array, m+1, r);

merge(array, l, m, r);

}

}

int main() {

cout<<"BHANU ARORA"<<endl;

int n;

cout << "Enter the number of elements: ";

cin >> n;

int arr[n];

cout << "Enter elements:" << endl;

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

cin >> arr[i];

}

cout << "Array before Sorting: ";

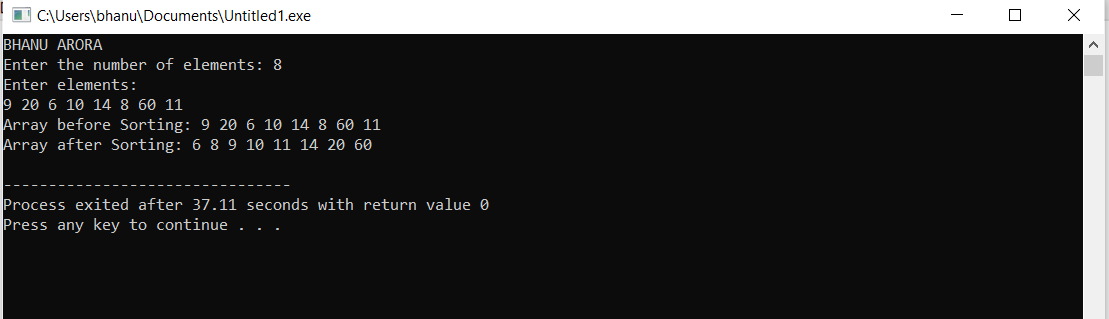
display(arr, n);

mergeSort(arr, 0, n-1);

cout << "Array after Sorting: ";

display(arr, n);

}



5.Quick

#include <bits/stdc++.h>

using namespace std;

// A utility function to swap two elements

void swap(int\* a, int\* b)

{

int t = \*a;

\*a = \*b;

\*b = t;

}

/\* This function takes last element as pivot, places

the pivot element at its correct position in sorted

array, and places all smaller (smaller than pivot)

to left of pivot and all greater elements to right

of pivot \*/

int partition (int arr[], int low, int high)

{

int pivot = arr[high]; // pivot

int i = (low - 1); // Index of smaller element and indicates the right position of pivot found so far

for (int j = low; j <= high - 1; j++)

{

// If current element is smaller than the pivot

if (arr[j] < pivot)

{

i++; // increment index of smaller element

swap(&arr[i], &arr[j]);

}

}

swap(&arr[i + 1], &arr[high]);

return (i + 1);

}

/\* The main function that implements QuickSort

arr[] --> Array to be sorted,

low --> Starting index,

high --> Ending index \*/

void quickSort(int arr[], int low, int high)

{

if (low < high)

{

/\* pi is partitioning index, arr[p] is now

at right place \*/

int pi = partition(arr, low, high);

// Separately sort elements before

// partition and after partition

quickSort(arr, low, pi - 1);

quickSort(arr, pi + 1, high);

}

}

/\* Function to print an array \*/

void printArray(int arr[], int size)

{

int i;

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

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

cout << endl;

}

// Driver Code

int main()

{

cout<<"BHANU ARORA"<<endl;

int arr[] = {9,20,6,10, 14, 8,60,11};

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

quickSort(arr, 0, n - 1);

cout << "Sorted array: \n";

printArray(arr, n);

return 0;

}

