**1.1 Bubble\_Sort\_2019450046.cpp**

#include<iostream>

using namespace std;

class arrays{

private:

int size, temp, arr[100], pass;

bool swap;

public:

void getdata(){

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

cin >> size;

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

cout << "Enter element no. " << i+1 << " : ";

cin >> arr[i];

}

cout << "Pass 0: ";

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

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

}

cout << endl;

}

void sortdata(){

pass=0;

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

swap = false;

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

if(arr[j]>arr[j+1]){

temp = arr[j];

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

arr[j+1] = temp;

pass++;

swap = true;

}

}

if(pass==0){

cout << "Array is already sorted.";

break;

}

else if(swap==true){

cout << "Pass " << i+1 << ": ";

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

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

cout << endl;

}

else

break;

}

cout << endl;

}

};

int main(){

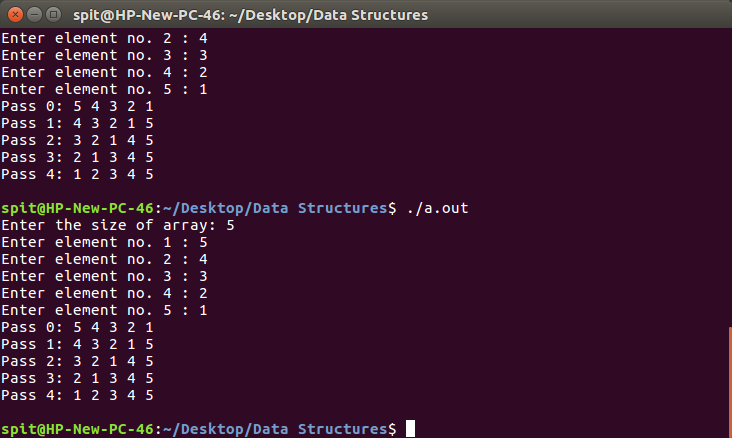
arrays ob;

ob.getdata();

ob.sortdata();

}

**Output:**



**1.2 Selection\_Sort\_2019450046.cpp**

#include<iostream>

using namespace std;

class arrays{

private:

int size, temp, arr[100], pass;

bool swap;

public:

void getdata(){

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

cin >> size;

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

cout << "Enter element no. " << i+1 << " : ";

cin >> arr[i];

}

cout << "Pass 0: ";

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

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

}

cout << endl;

pass=0;

}

void selection\_sort(){

int max;

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

max = size - i - 1;

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

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

max = j;

}

}

if(max!=size-i-1){

temp = arr[size-i-1];

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

arr[max] = temp;

cout << "Pass " << ++pass << ": ";

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

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

cout << endl;

}

}

cout << endl;

}

};

int main(){

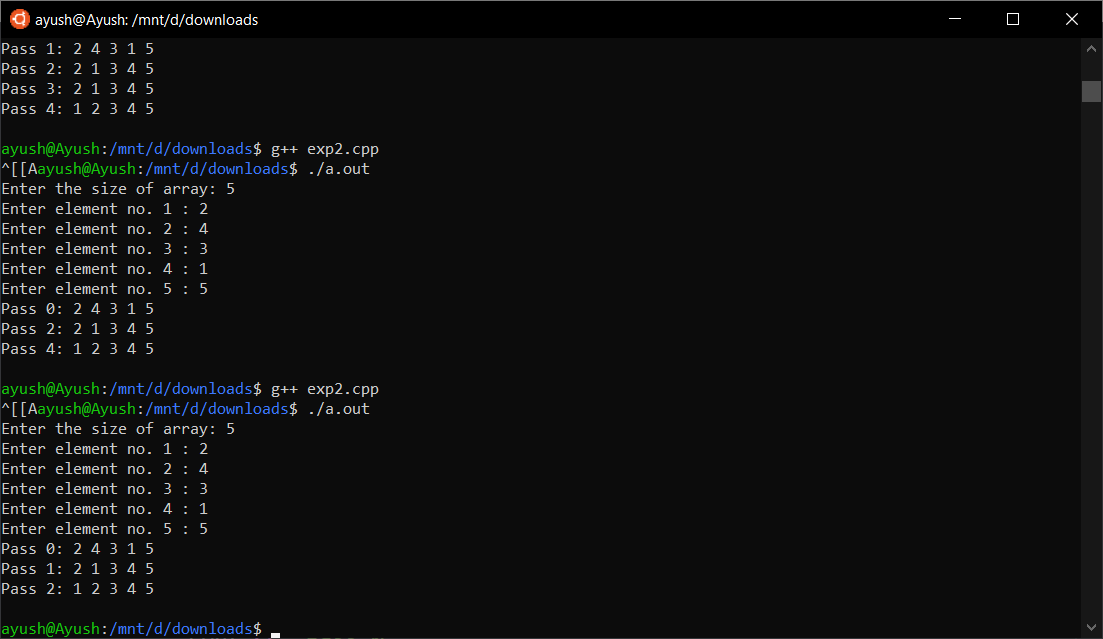
arrays ob;

ob.getdata();

ob.selection\_sort();

}

**Output:**



**1.3 Insertion\_Sort.cpp**

#include<iostream>

using namespace std;

class arrays{

private:

int size, temp, arr[100], pass;

bool swap;

public:

void getdata(){

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

cin >> size;

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

cout << "Enter element no. " << i+1 << " : ";

cin >> arr[i];

}

cout << "Pass 0: ";

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

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

}

cout << endl;

pass=0;

}

void insertion\_sort(){

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

swap = false;

temp = arr[i];

int j = i-1;

while(j>=0 && arr[j]>temp){

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

j--;

swap = true;

}

if(swap == true){

arr[j+1]=temp;

cout << "Pass " << ++pass << ": ";

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

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

cout << endl;

}

}

cout << endl;

}

};

int main(){

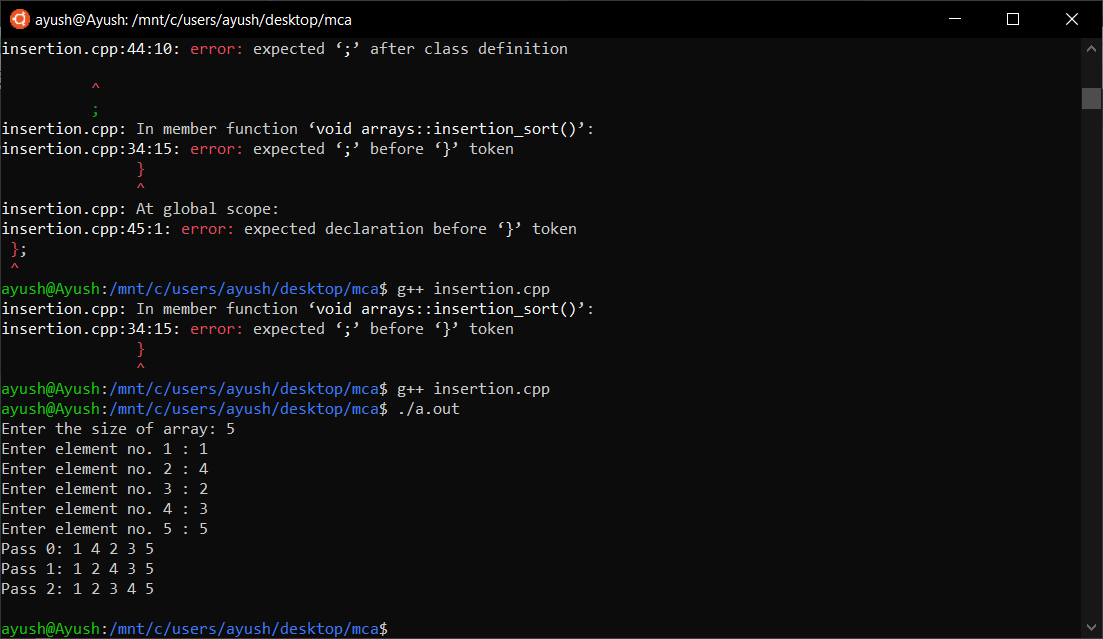
arrays ob;

ob.getdata();

ob.insertion\_sort();

}

**Output:**



**1.4 Radix\_Sort.cpp**

#include<iostream>

using namespace std;

class arrays{

private:

int size, temp, arr[100], pass, max;

bool swap;

public:

void getdata(){

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

cin >> size;

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

cout << "Enter element no. " << i+1 << " : ";

cin >> arr[i];

}

pass = -1;

cout << endl;

display();

}

void display(){

cout << "Pass " << ++pass << ": ";

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

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

}

cout << endl;

}

int get\_max(){

max = arr[0];

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

if (max < arr[i])

max = arr[i];

}

}

void radix\_sort(){

get\_max();

for(int pos = 1; max/pos>0; pos\*=10){

count\_sort(pos);

}

}

void count\_sort(int pos){

int temp\_arr[size];

int freq[10] = {0};

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

++freq[(arr[i]/pos)%10];

for(int i = 1; i <= 9; i++)

freq[i] += freq[i-1];

for(int i = size - 1; i >= 0; i--)

temp\_arr[--freq[(arr[i]/pos)%10]] = arr[i];

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

arr[i] = temp\_arr[i];

display();

}

};

int main(){

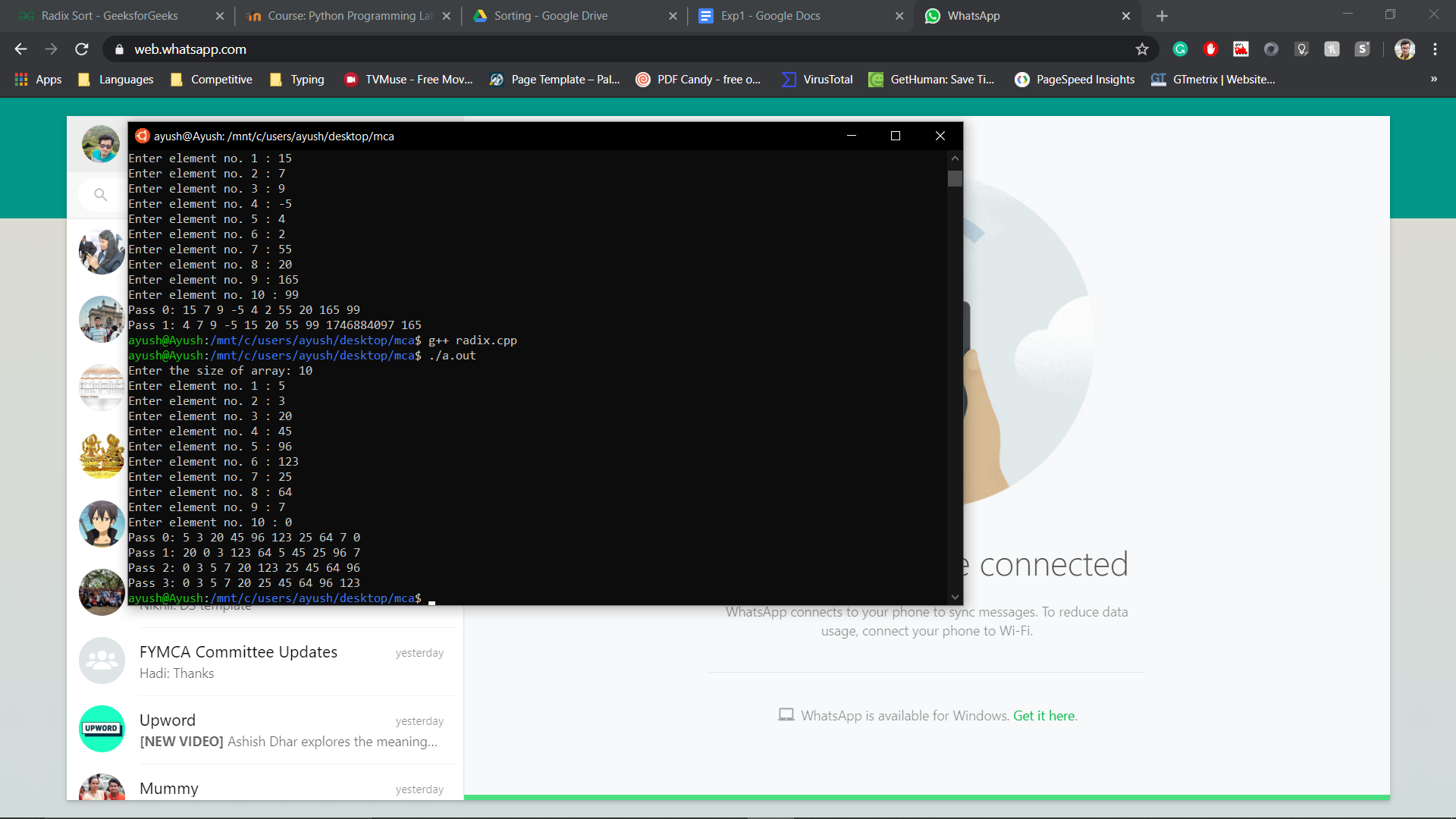
arrays ob;

ob.getdata();

ob.radix\_sort();

}

**Output:**



**1.5 Shell\_Sort.cpp**

#include<iostream>

using namespace std;

class arrays{

private:

int size, temp, arr[100], pass, max;

public:

void getdata(){

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

cin >> size;

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

cout << "Enter element no. " << i+1 << " : ";

cin >> arr[i];

}

pass = -1;

cout << endl;

display();

}

void display(){

cout << "Pass " << ++pass << ": ";

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

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

}

cout << endl;

}

void shell\_sort(){

for(int i = size/2; i >= 1; i/=2){

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

for(int k = j - i; k >= 0; k-=i){

if(arr[k+i]>arr[k]){

break;

}

else{

temp = arr[k+i];

arr[k+i] = arr[k];

arr[k] = temp;

}

}

}

display();

}

}

};

int main(){

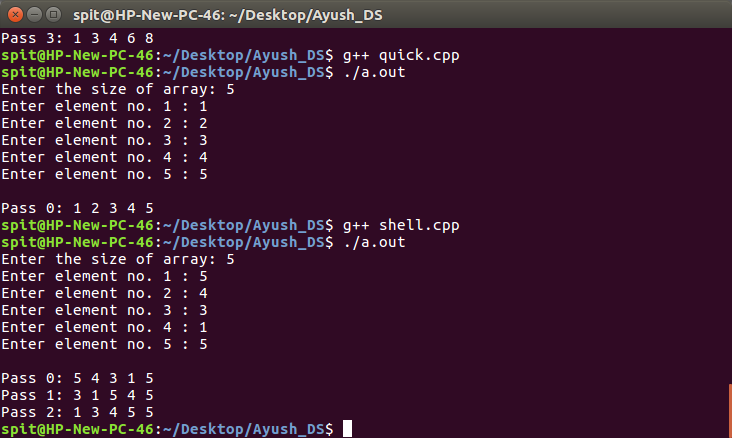
arrays ob;

ob.getdata();

ob.shell\_sort();

}

**Output:**



**1.6 Quick\_Sort.cpp**

#include<iostream>

using namespace std;

class arrays{

private:

int size, temp, arr[100], pass, lock;

bool swapping;

public:

void getdata(){

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

cin >> size;

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

cout << "Enter element no. " << i+1 << " : ";

cin >> arr[i];

}

pass = -1;

cout << endl;

display();

quick\_sort(0 , size-1);

}

void display(){

cout << "Pass " << ++pass << ": ";

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

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

}

cout << endl;

}

void quick\_sort(int start, int end){

if(start < end){

lock = partition(start, end);

quick\_sort(start, lock-1);

quick\_sort(lock+1, end);

}

}

int partition(int start,int end){

int pivot = arr[start];

int i = start;

int j = end;

while(i<j){

while(arr[i] <= pivot)

i++;

while(arr[j] > pivot)

j--;

if(i < j){

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

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

temp = arr[start];

arr[start] = arr[j];

arr[j] = temp;

display();

}

return j;

}

};

int main(){

arrays ob;

ob.getdata();

}

**Output:**

