**EXPERIMENT-3**

**AIM:** Insertion Sort with its worst and best case analysis.

**Code:-**

#include <iostream>

using namespace std;

void insertion\_sort(int A[], int size, int \*analyser){

    int i,j, x;

    for(i = 1; i<size; i++){

        j = i-1;

        x = A[i];

        while(j>-1 && A[j]>x){

            A[j+1] = A[j];

            j--;

            (\*analyser)++;

        }

        A[j+1]=x;

    }

}

void display\_elements\_arr(int A[], int x)

{

    cout << "| ";

    for (int i = 0; i < x; i++)

    {

        cout << A[i] << " | ";

    }

    cout << endl;

}

void fill\_array(int A[], int size){

    int i;

    for(int i =0; i<size; i++){

        cout<<"Enter data at position "<<i<<" : ";

        cin>>A[i];

    }

}

int main(){

    int size, analyser = 0;

    cout<<"Enter size: ";

    cin>>size;

    int \*A = new int(size);

    fill\_array(A, size);

    display\_elements\_arr(A, size);

    insertion\_sort(A, size, &analyser);

    display\_elements\_arr(A, size);

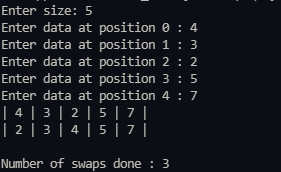
    cout<<"\nNumber of swaps done : "<<analyser<<endl;

    return 0;

}

**GENERAL CASE :-**

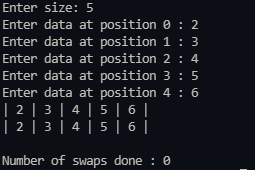
Lets take array of size 5 and its element as 4,3,2,5,7 then



Number of swaps done is 3.

**BEST CASE :-**

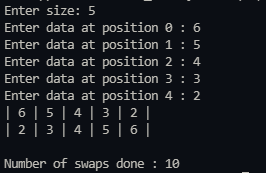
When the elements of the array is already in sorted manner.



Then number of swaps done is 0.

**WORST CASE :-**

When none of the elements of the array is in sorted manner.



Then number of swaps done is 10.

i.e. size\*(size-1)/2