1. **Code and analysis of Iterative based sorting algorithm.**

#include<stdio.h>

int count = 0;

void bubbleSort(int A[],int n){

int i,j,temp;

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

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

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

temp = A[j+1];

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

A[j] = temp;

}

count+=7;

}

count+=4;

}

}

void selectionSort(int A[],int n){

int i,j,least,loc;

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

least = A[i];

loc = i;

for(j=i;j<n;j++){

if(A[j]<least){

least = A[j];

loc = j;

}

count+=6;

}

A[loc] = A[i];

A[i] = least;

count+=7;

}

}

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

int i,j,key;

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

key = A[i];

for(j=i-1;j>=0&&A[j]>key;j--){

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

count+=5;

}

A[j+1] = key;

count+=6;

}

}

void display(int A[],int n){

int i;

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

printf("%d\t",A[i]);

}

}

int main(){

int A[] = {2,4,1,5,87,6};

int n = sizeof(A)/sizeof(int);

int op;

printf("The unsorted array is: \n");

display(A,n);

printf("\n\n1- For Bubble sort \n2- For Selection sort \n3- For Insertion Sort\n");

printf("Enter the option: ");

scanf("%d",&op);

switch(op){

case 1:

bubbleSort(A,n);

break;

case 2:

selectionSort(A,n);

break;

case 3:

insertion\_sort(A,n);

break;

default:

printf("\nInvalid option...");

}

printf("\nThe sorted array is: \n");

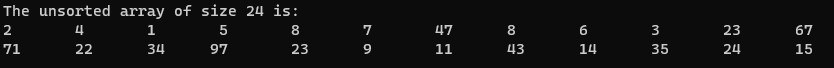
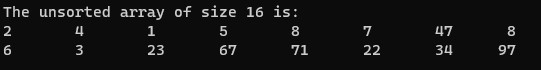
display(A,n);

printf("\nNumber of steps to complete sort: %d",count);

return 0;

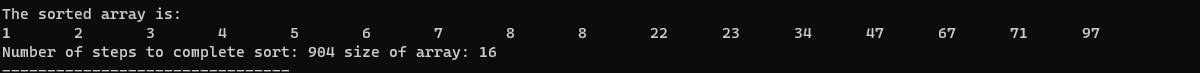
}

**Output:** Unsorted array are:

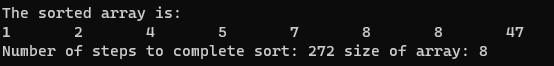
****

Sorted array using following algorithm:

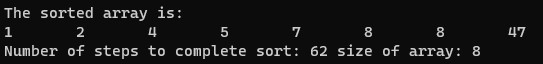
1. **Bubble Sort**

****

1. **Selection Sort**

****

1. **Insertion Sort**

****

**Comparison between Iterative Sorting algorithms**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bubble sort | Selection Sort | Insertion Sort |
| 8 | 228 | 272 | 62 |
| 16 | 904 | 928 | 205 |
| 24 | 2028 | 1968 | 538 |