**UNIVERSITY OF DELHI**

**ATMA RAM SANATAN DHARMA COLLEGE**

**DS ASSIGNMENT**

**SUBMITTED BY -**

**SUYASH PRATAP MALL**

**( 88051 )**

**Q.** Write a Program to generate the Fibonacci Series using recursion.

**ANS**-

#include<iostream>

using namespace std;

int fibo(int); // To find fibonacci sequence (recursively)

void fibonaciiSeries(int); // To print fibonacci Series

int fibo(int num ){

if (num==1){

return 1;

}

else if(num==0){

return 0;

}

else{

return fibo(num-1) + fibo(num-2);

}

}

void fibonaciiSeries(int n){

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

cout<<fibo(i)<<" ";

}

}

int main(){

int i;

cout<<"Enter the no. of terms want to print(of fibonacci series) : ";

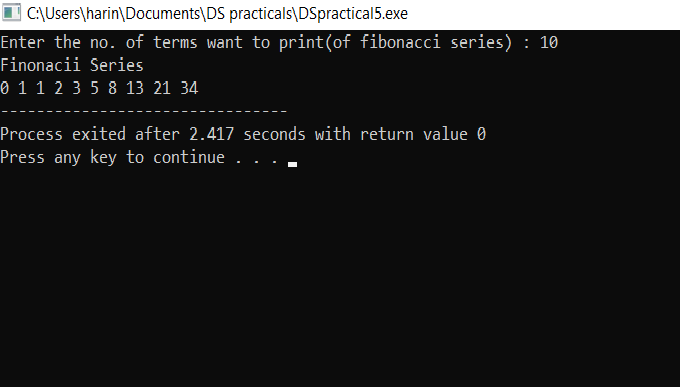
cin>>i;

cout<<"Finonacii Series"<<endl;

fibonaciiSeries(i);

return 0;

}



**Q.** Write a Program to implement Tower of Hanoi using recursion.

**ANS-**

#include<iostream>

using namespace std;

void towerOfHanoi(int n,char src,char dest,char helper){

if(n<=0){

return;

}

towerOfHanoi(n-1,src,helper,dest);

cout<<"Move from "<<src<<" to "<<dest<<endl;

towerOfHanoi(n-1,helper,dest,src);

}

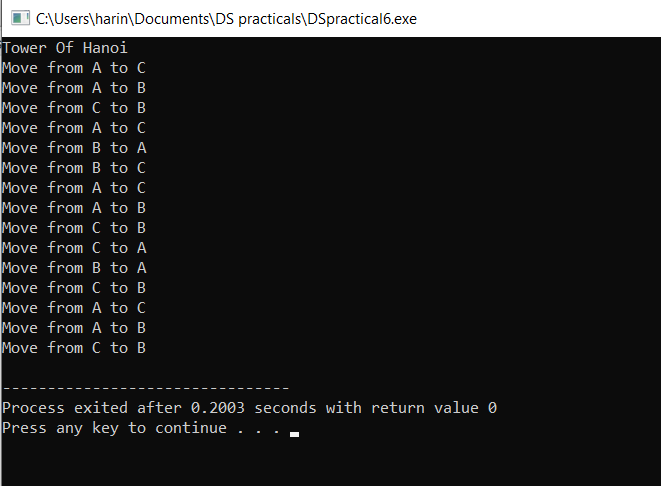
int main(){

cout<<"Tower Of Hanoi"<<endl;

towerOfHanoi(4,'A','B','C');

return 0;

}



**Q. Write a Program to implement binary search using recursion.**

**ANS -**

#include<iostream>

#include<string>

using namespace std;

void displayArray(int\* &array , int\* &size){

cout<<endl;

cout<<"[ ";

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

cout<<array[i];

if(i!= \*size-1){

cout<<" ,";

}

}

cout<<" ]";

cout<<endl;

}

string binarySearch(int arr[] , int key,int start,int end){

if(key<arr[start] && key>arr[end-1]){

return "Sorry, Element isn't present";

}

int mid = (start+end)/2;

if(arr[mid]==key){

return "Element found at position '"+to\_string(mid+1)+"'";

}

else if (key>arr[mid])

{

return binarySearch(arr , key,mid+1,end);

}

else{

return binarySearch(arr , key,start,mid-1);

}

}

int main()

{

int arr[] = {1,3,4,5,8,21,34,35,44,56,60,67};

int key;

int \*size = new int(12);

int \*array = arr;

cout<<"Enter the element to search : ";

cin>>key;

cout<<endl;

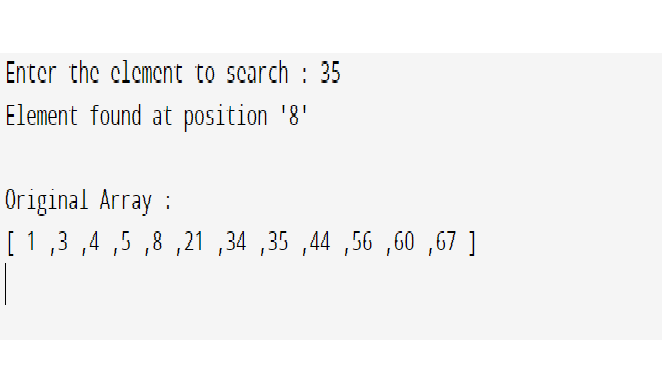
cout<<binarySearch(arr ,key ,0,12)<<endl<<endl;

cout<<"Original Array : ";

displayArray(array,size);

return 0;

}



**Q. Write a Program to implement Bubble Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of algorithm.**

**ANS-**

#include<iostream>

using namespace std;

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

{

int temp =\*a;

\*a=\*b;

\*b=temp;

}

void bubbleSort(int arr[],int n)

{

int noOfSwaps;

int noOfComparisons;

bool swapped;

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

{

noOfSwaps=0;

noOfComparisons=0;

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

{

noOfComparisons++;

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

{

swap(&arr[j],&arr[j+1]);

noOfSwaps++;

}

}

std::cout<<"Array after"<<i+1<<" iteration: \n";

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

std::cout<<arr[k]<<" ";

}

std::cout<<"\nNo of swaps: "<<noOfSwaps;

std::cout<<"\nNo of comparisons: "<<noOfComparisons<<std::endl<<std::endl;

}

}

void printArray(int arr[],int size)

{

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

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

std::cout<<"\n";

}

int main()

{

int arr[]={64,34,25,12,22,11};

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

std::cout<<"Unsorted array:\n";

printArray(arr,n);

cout<<endl;

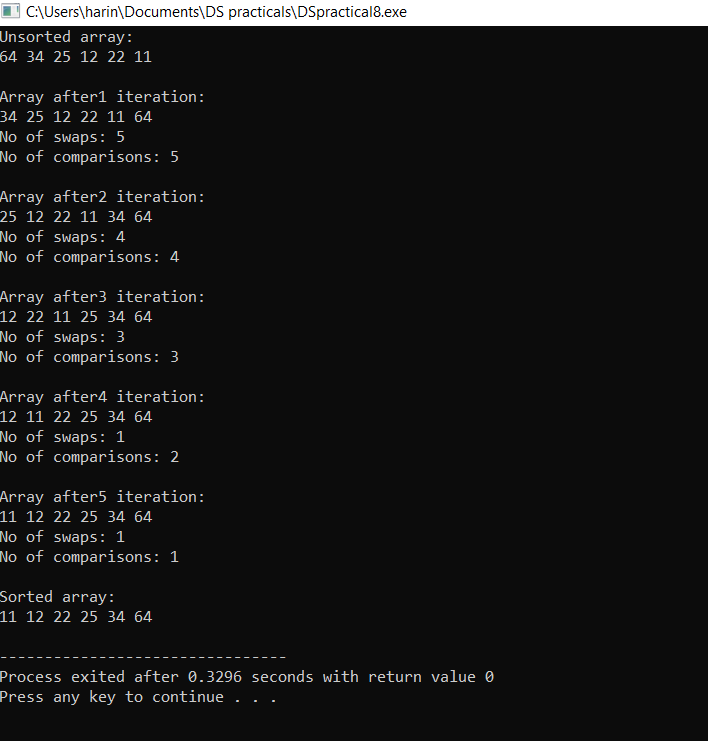
bubbleSort(arr,n);

std::cout<<"Sorted array:\n";

printArray(arr,n);

return 0;

}



**Q. Write a Program to implement Insertion Sort. Find the number of comparisons during each pass and display the intermediate result. Use the observed values to plot a graph to analyse the complexity of algorithm.**

**ANS-**

#include<iostream>

using namespace std;

void displayArray(int\* &array , int\* &size){

cout<<"\n";

cout<<"[ ";

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

cout<<array[i];

if(i!= \*size-1){

cout<<" ,";

}

}

cout<<" ]"<<endl<<endl;

}

void insertionSort(int\* &array , int\* &size){

int key;

for (int i = 1; i <\*size ; i++)

{

key = array[i];

int j = i-1;

int\* comparisons = new int(0);

while (key<array[j] && j>=0){

\*comparisons+=1;

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

j--;

}

array[j+1]=key;

cout<<"No. of Iterations : "<<i<<endl;

cout<<"No. of Comparisons : "<<\*comparisons<<endl;

}

}

int main(){

int arr[] = {7,38,3,60,76,8,3,10,70,8,19,31};

int \*array = arr;

int \*length = new int(12);

cout<<endl<<"UNSORTED";

displayArray(array,length);

insertionSort(array,length);

cout<<endl<<"SORTED";

displayArray(array,length);

delete[] array;

delete length;

return 0;

}

