**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-112L Data Structure** **& Algorithm Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 4th**

**Submission Date: 26th April 2022**

**Submitted to: Engr. Rafi-Ullah**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Signature**

**Lab No. 03**

**Recursion & Searching Algorithms**

**Objectives:**

In this lab we will be discussing about Recursion & Searching Algorithms in detail. This is one of the most important concepts in C++ language.

* To understand & implement the working of recursive functions in C++.
* To understand & Implement searching algorithms in C++.

A function that calls itself is known as recursive function. And this technique is known as recursion. The figure below shows how recursion works by calling itself repeatedly.

The recursion continues until some condition is met. To prevent infinite recursion, if...else statement (or similar approach) can be used where one branch makes the recursive call and other doesn't.

**Tools/Software Required:**

* All the tasks are implemented on DEV C++.

**Introduction:**

**LINEAR SEARCH:**

The simplest form of a search is the linear search. This technique is meant for searching a particular item in an unsorted data set in a sequential manner until the desired data item is found. Linear search is easy to write and efficient for short lists, but inefficient for long ones. To find any element in a long array (list), there are far more efficient methods, provided that the array is sorted.

**BINARY SEARCH:**

* Binary search is a simple method of accessing a particular item in a sorted (ordered) data set. A search for a particular item with a certain key value resembles the search for a name in telephone directory or a word in a dictionary. The approximate middle item of the data set is located, and its key value is examined. If its value is too high, then the key to the middle element of the first half of the set is examined and procedure is repeated on the first half until the required item is found. If the value is too low, then the key to the middle entry of the second half of the data set is tried and the procedure is repeated on the second half. This process continues until the desired key is found or search interval becomes empty. The binary search algorithm is based on binary search tree. A queue is different from a stack in that a queue works on a First in First out (FIFO) basis. In general, items are added to the end of a queue (In the way that we join at the end of a queue for a bus), and items are removed from the front of a queue. (The people at the front of the queue for the bus can get on the bus first.)

**Lab Tasks:**

**Task 1:**

Write a program to find out a number among all other numbers entered by user using linear search technique.

**Code:**

**//RECURSIVE LINEAR SEARCH**

**#include <iostream>**

**using namespace std;**

**void display(int arr[],int nsize);**

**int recursiveLinearSearch(int arr[],int num,int nsize);**

**int main()**

**{**

**int num,nsize;**

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

**cin>>nsize;**

**int arr[nsize];**

**for(int i=0; i<nsize; i++)**

**{**

**cout<<"Enter number at index arr["<<i<<"] ";**

**cin>>arr[i];**

**cout<<endl;**

**}**

**display(arr, nsize);**

**cout << "Enter the number you want to find: ";**

**cin >> num;**

**int a= recursiveLinearSearch(arr, num ,nsize);**

**cout << "Number is found at location " << a;**

**return 0;**

**}**

**// Recursive**

**int recursiveLinearSearch(int arr[],int num,int nsize)**

**{**

**nsize=nsize-1;**

**if(nsize <0) {**

**return -1;**

**}**

**else if(arr[nsize]==num) {**

**return nsize+1;**

**}**

**else {**

**return recursiveLinearSearch(arr,num,nsize);**

**}**

**}**

**// Display**

**void display(int arr[],int nsize)**

**{**

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

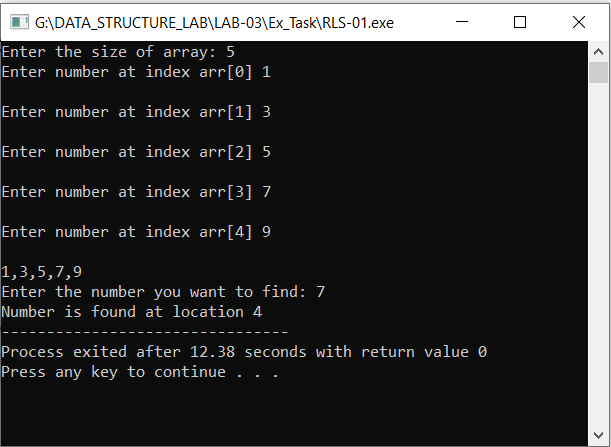
**cout<<arr[i]<<",";**

**}**

**cout<<"\b \b"<<endl;**

**}**

**Output:**

****

**Task # 02:**

Write a program to find out a number among all other numbers entered by user using Binary search technique.

**Code:**

**#include <iostream>**

**using namespace std;**

**int binarySearch(int arr[], int start, int end, int num);**

**void display(int arr[],int size);**

**int main(void)**

**{**

**int size, num;**

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

**cin>>size;**

**int arr[size];**

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

**{**

**cout<<"Enter number at index arr["<<i<<"] ";**

**cin>>arr[i];**

**cout<<endl;**

**}**

**display(arr, size);**

**cout << "\nEnter the number you want to find: ";**

**cin >>num;**

**int end=size-1;**

**int result = binarySearch(arr, 0, end, num);**

**if(result == -1)**

**{**

**cout<<"Element is not present in array"<<endl;**

**}**

**else**

**{**

**cout<<"Element is present at index "<<result;**

**}**

**return 0;**

**}**

**void display(int arr[],int nsize)**

**{**

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

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

**}**

**cout<<"\b \b"<<endl;**

**}**

**int binarySearch(int arr[], int start, int end, int num)**

**{**

**while (start <= end)**

**{**

**int mid = (start + end)/2;**

**if (arr[mid] == num)**

**return mid;**

**else if (arr[mid] < num)**

**start = mid + 1;**

**else**

**end = mid - 1;**

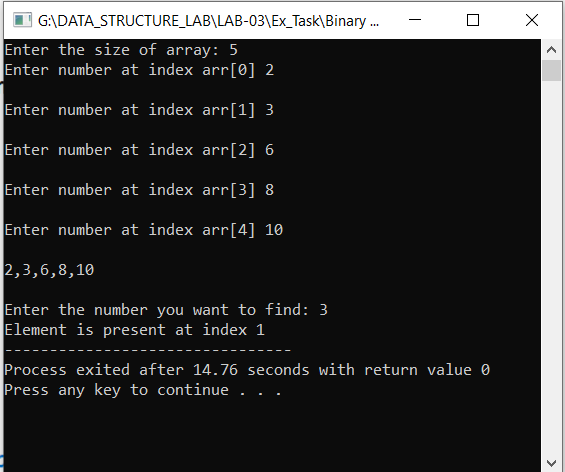
**}**

**// if we reach here, then element was not present**

**return -1;**

**}**

**Output:**

****

**Task # 03:**

Find Sum of Fibonacci Series using Recursive Function.

**Code:**

**//Fibonacci Series using Recursion**

**#include <iostream>**

**using namespace std;**

**int fib(int x)**

**{**

**if((x==1)||(x==0))**

**{**

**return(x);**

**}**

**else {**

**return(fib(x-1)+fib(x-2));**

**}**

**}**

**int main()**

**{**

**int x , i=0;**

**cout << "Enter the number of terms of series : ";**

**cin >> x;**

**cout << "\nFibonnaci Series : ";**

**while(i < x)**

**{**

**cout << " " << fib(i);**

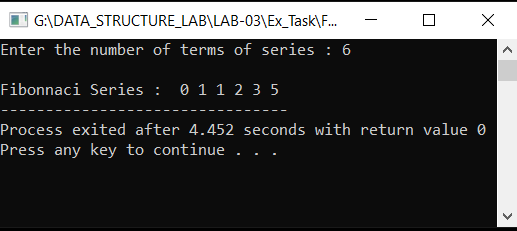
**i++;**

**}**

**return 0;**

**}**

**Output:**

****

**Results & Observations:**

In this Lab I've learned about the concept of Recursion & Searching Algorithms & also understand the difference between Linear Search & Binary Search. Further I understand the concept of Fibonacci Series using Recursive Function. In the first task, I've used Linear Search Recursive function that will call itself until the required condition don’t get false. In the second task, I've used Binary Search Recursive function in which I've used while loop to search the required number to the half of the required sorted array. In the Third task, I've used Fibonacci function that’s also recursive until the number reaches to 0 or 1, it’ll call itself to perform addition of numbers in sequence wise.