Name : Dhanraj Kore

Roll no : 60

Div : B Batch : B-3

1. Fibonacci Search :

Code:

#include<iostream>

using namespace std;

void FibonacciSearch(int \*a, int start, int end, int \*fab, int index, int item)

{

    int i, mid;

    // Assigning middle of the array using Fibonacci element.

    mid = start+fab[index-2];

    // Return if item found at mid index.

    if(item == a[mid])

    {

        cout<<"\n item found at "<<mid<<" index.";

        return;

    }

    // Return if item found at start index.

    else if(item == a[start])

    {

        cout<<"\n item found at "<<start<<" index.";

        return;

    }

    // Return if item found at end index.

    else if(item == a[end])

    {

        cout<<"\n item found at "<<end<<" index.";

        return;

    }

    // If mid becomes start or end of the sub-array then element not found.

    else if(mid == start || mid == end)

    {

        cout<<"\nElement not found";

        return;

    }

    // According to the item value choose the partion to proceed further.

    else if(item > a[mid])

        FibonacciSearch(a, mid, end, fab, index-1, item);

    else

        FibonacciSearch(a, start, mid, fab, index-2, item);

}

main()

{

    int n, i, biter, fab[20], a[20]={1, 9, 18, 24, 27, 35, 38, 41, 49, 53, 55, 66, 67, 72, 75, 77, 81, 89, 90, 97};

    char ch;

    fab[0] = 0;

    fab[1] = 1;

    i = 1;

    while(fab[i] < 20)

    {

        i++;

        fab[i] = fab[i-1]+fab[i-2];

    }

    cout<<"\nEnter the Element to be searched: ";

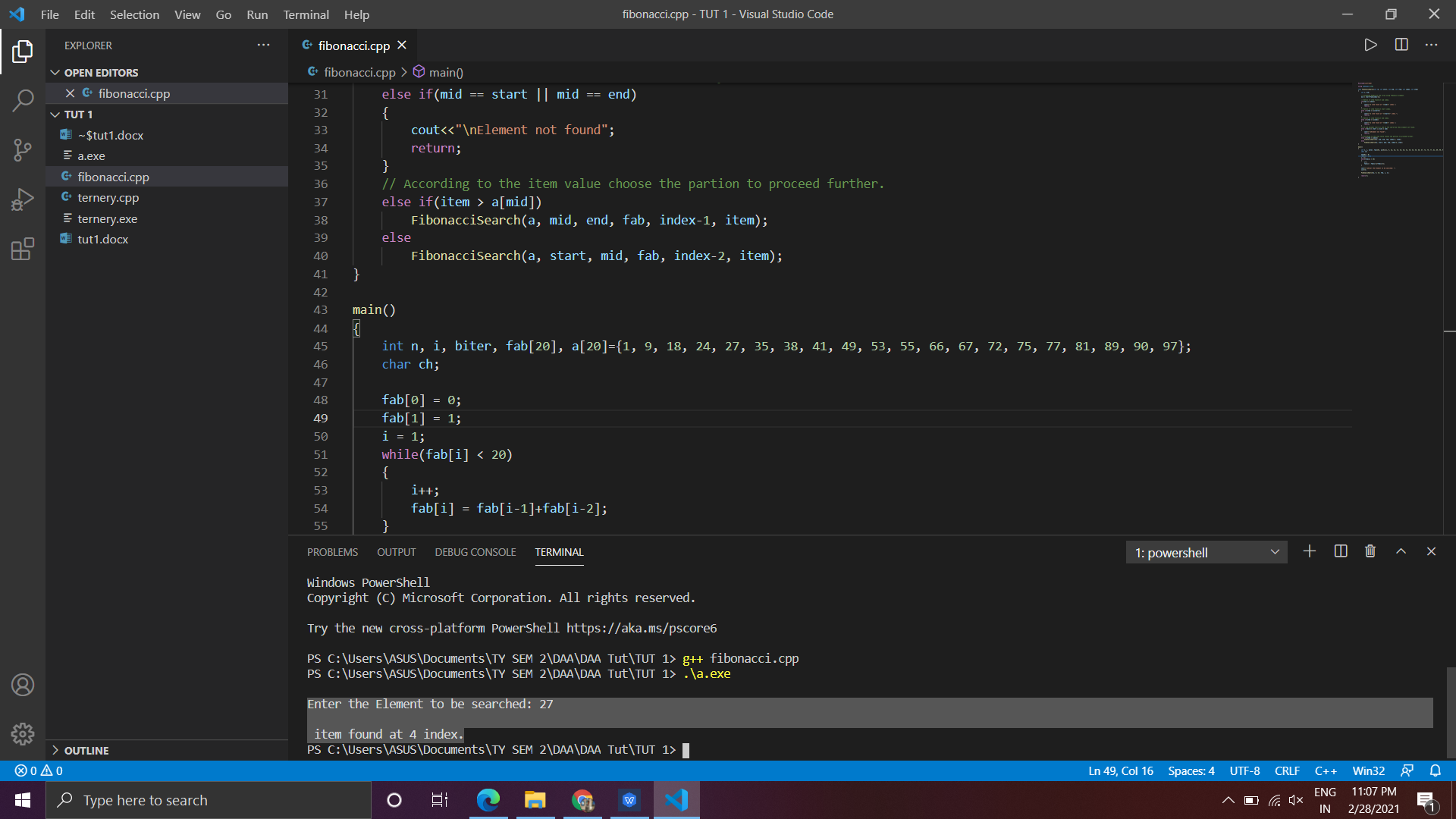
    cin>>n;

    FibonacciSearch(a, 0, 19, fab, i, n);

    return 0;

}

O/P:



1. Ternary Search:

Code:

#include <iostream>

using namespace std;

int ternarySearch(int l, int r, int key, int ar[])

{

    while (r >= l)

    {

        int mid1 = l + (r - l) / 3;

        int mid2 = r - (r - l) / 3;

        if (ar[mid1] == key)

        {

            return mid1;

        }

        if (ar[mid2] == key)

        {

            return mid2;

        }

        if (key < ar[mid1])

        {

            r = mid1 - 1;

        }

        else if (key > ar[mid2])

        {

            l = mid2 + 1;

        }

        else

        {

            l = mid1 + 1;

            r = mid2 - 1;

        }

    }

    return -1;

}

int main()

{

    int l, r, p, key;

    int ar[100], n;

    cout << "Enter no of elements : ";

    cin >> n;

    cout << "Enter array elements : " << endl;

    for (int i = 0; i < n; i++)

        cin >> ar[i];

    cout << "Enter the element to be searched : ";

    cin >> key;

    l = 0;

    r = n;

    p = ternarySearch(l, r, key, ar);

    if(p!=-1)

    cout << "Index of " << key << " is " << p << endl;

    else

    cout<<"Element "<<key<<" not found"<<endl;

}

O/P:

