10/18/2021

Department Of Computer Science

Subject: Data Structure and Algorithm **Instructor:** Ma'am Zainab Malik

Lab No: 2 **Date:** 18-10-2021

Class: BSCS-3B

Students' Name

1) Madina Javed Iqbal Khokhar 2426
2) Esha Mansoor 2413
3) Sultan Zahid 2411
4) Abdul Moeed 2419

Lab Repot 2

Task 1:

Implement a program that will traverse the array and while traversing it will add the values of that array.

Description:

To traverse an array means to access each element (item) stored in the array so that the data can be checked or used as part of a process.

In order to transverse an array and to add values in it, firstly we will declare a function names Traverse in which we will define to traverse elements entered by the user and then to add them one by one.In the main function, we will ask a user to enter the size of an array then we will take values from user for an array using loop. Lastly, we will call our function to perform our task.

Code:

```
#include <iostream>

using namespace std;

void Traverse(int array[], int n);          // Function Declaration

int main()

{

        int n=0;

        cout<< "Enter size of Array: ";

        cin>>n;                             // Take input from user

        int arr[n];

        cout<<"Enter values at indexes form 0 to "<<n-1<<endl;

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

        {

                cin>>arr[i];

        }

        Traverse(arr,n);                // Function Call

}

void Traverse(int array[], int n)       // function Defination

{
```

```
Int sum=0;
        for (int i=0; i<n; i++)
        {
                cout<<array[i]<<"\t";
cout<<endl;

        sum=sum+array[i];                    // Add the value of array
        }
Cout<<"Sum of Array="<<sum;        // Display the sum of that array
}
```

Output:

```
Enter size of Array: 5
Enter values at indexes form 0 to 4
4
5
3
2
2
Traversing Array
4
5
3
2
2
sum of array=16

--------------------------------
```

Task 2:

Implement a program that will whether a entered matrix is a identity matrix or not.

Description:

Firstly, we will declare a function named identity in which we will define it to display the matrix entered by the user using a for loop, then again using for loop we will check our condition in order to find out whether our matrix is identity matrix or not. Within the for loop, we used if statement to check whether the diagonal items are ones and the non-diagonal items are zeros. If it is true, we changed the Flag value to 1 and apply the break statement to exit from the for loop. Finally, we will use If Else statement to print the identity matrix output based on the Flag value.

Code:

```
#include<iostream>

using namespace std;

void identity(int[3][3]);                // Function Declaration

int main(){

   int a[3][3];                          // Declare 2d array

   cout<<"Enter Elements: ";

   for(int i = 0; i<3 ; i++)

     {                                   // Start of outer loop

     for(int y=0; y<3 ; y++)

       {                                 Start of inner loop

      cin>>a[i][y];                     // Taking input from the user i.e(elements of array, rows and columns)

   }                                    // End of inner loop

 }                                      // End of outer loop

   identity(a);                         // Function Call

}

void identity(int arr[3][3])      Function Defination

{                                       //Start of function Defination

   int flag = 0;

   for(int i = 0; i<3; i++)

   {                                    // Start of outer loop

      for(int y=0; y<3; y++)
```

```
    {                                   // Start of inner loop
        cout<<arr[i][y];        // Display the array which we entered


        }                               // End of inner loop
    cout<<endl;
}                                   // End of outer loop
  for(int i = 0; i<3; i++)
  {                                     // Start of outer loop
    for(int y=0; y<3; y++)
    {                                       // Start of inner loop
        if(i==y && arr[i][y]!=1)
        {                                   // Start of if loop
        flag = 1;
            break;
    }                                   // End of if loop
        else if(i!=y && arr[i][y]!=0)
        {                                   // Start of else if loop
        flag = 1;
            break;
    }                                   // End of else if loop
        }                               // End of inner loop
}                                   // End of outer loop
  if(flag == 1)
  {                                 // Start of if loop
  cout<<"Matrix is not identity"<<endl;
}                                   //End of if loop
  else
  {                                 // Start of else loop
    cout<<"Matrix is identity"<<endl;
  }                                 // End of else loop


}                   // End of Function Defination
```

Output

```
enter row=3
enter column=3
enter the matrix elements=1 0 0
0 1 0
0 0 1
Matrix is Identity matrix
--------------------------------
Process exited after 10.68 seconds with return value 0
Press any key to continue . . .
```

```
Enter Elements: 0 1 0
0 0 1
1 0 0
010
001
100
Matrix is not identity

--------------------------------
Process exited after 9.625 seconds with return value 0
Press any key to continue . . .
```

Task 3

Implement linear search algorithm in array as function.

Description:

Linear search is a sequential searching algorithm where we start from one end and check every element of the list until the desired element is found. It is the simplest searching algorithm.

In this program we will declare a function named linearSearch in which we will define a for loop use a if statement to compare the entered value by the user to all the elements in an array. If that particular values is found we display a message “Item found” and return its index number ,however, if our required element is not found we wil display a message “Item not found” and return -1 which is not possible.

In the main function we will ask the user to type the size of an array and will take values form the user of an array using a for loop. Finally, we call our function , to perform our required task.

Code:

```
#include <iostream>

using namespace std;

int linearSearch(int array[], int n, int item);            // Function Declaration

int main()

{

        int n=0;

        int item;                                   // Declaring item;

        cout<< "Enter size of Array: ";

        cin>>n;                                     // Taking inpit from user regarding size of array

        int arr[n];                                 // Declaring array;

        cout<<"Enter values at indexes form 0 to "<<n-1<<endl;

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

        {                                           //Start of for loop

                cin>>arr[i];                        // Taking input from user

        }                                           // End of for loop


        cout<<"\nEnter Value you want to search: ";

        cin>>item;                          // Taking item from the user which he wants to search

        cout<<linearSearch(arr,n,item);     // Function Call
```

```
}                                       // End of main



Int linearSearch(int array[], int n, int item)     //Function Defination
{                                       // Start of Function Defination
        for (int i=0; i<n; i++)
        {                               // Start of  for loop
                if (array[i]==item)
                {                       // Start of if loop
                        cout<<"item found"<<endl;
                        return i;       // Returns index number
                }                       // End of if loop
        }                               // End of for loop
        cout<<"item not found"<<endl;
        return -1;
}                                       // End of Function Defination
```

Output

```
Enter size of Array: 4
Enter values at indexes form 0 to 3
2
3
5
6

Enter Value you want to search: 5
item found
2
--------------------------------
Process exited after 11.86 seconds with return value 0
Press any key to continue . . .
```

Task 4:

Implement binary search algorithm in array as function.

Description:

Key points to keep in mind:

1) Binary search can be implemented only on a sorted list of items. If the elements are not sorted already, we need to sort them first.
2) The search starts with comparing the target element with the middle element of the array. If value matches then the position of the element is returned.
3) In case the target element is less than the middle element (considering the array follows an ascending order) of the array then the second half of the array is discarded and the search continues by dividing the first half.
4) The process is the same when the target element is greater than the middle element, only, in this case, the first half of the array is discarded before continuing with the search. The iteration repeats until a match for the target element is found.

The given below program is illustrating the usage binary search algorithm in which we have build a function named binarySearch where we have use a while loop and if else statement to compare a search key and middle term.

In the main function ,we have declare an array of size ten and then ask the user to enter the element he/she want to search . After calling our function, we have use a if else statement in which we declare if value of index is -1(location is not possible) display a message “value is not founded” else value is found at a particular index (message) will be displayed.

Code:

```
#include <iostream>

using namespace std;

 int binarySearch(int arr[], int beg, int end, int item);       // Function Declaration

int main()

{                                                   // Start of main

  int array[10] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};       // Declaring Array

  int item = 0;                                             // Declaring item

cout<<array[10]<<endl;

  cout << "Enter Search Element: ";
```

```
  cin >> item;                                    // Taking the item from user
  int index;                                      Declaring index
  int size;                                       Declaring size
  size = sizeof(array)/sizeof(array[0]);
  index=binarySearch(array,0,size,item);      // Function Call
   if(index == -1)
    cout << item << " Not Found" << endl;
  else
    cout << item << " Found at Index = " << index << endl;

  return 0;
}                                                 // End of main
  int binarySearch(int arr[], int beg, int end, int item) {       // Function Defination
  int mid = 0;                                    // Declaring mid
  while(beg < end) {
    mid = (beg+end) / 2;              // find middle index
    if(item > arr[mid])          // compare search key and middle term
      beg = mid + 1;
    else
      end = mid;
  }                                       // End of while
  if(item == arr[beg])
    return beg; // key found

  return -1; // key not found
}                                     // End of Function Defination
```

```
16
Enter Search Element: 50
50 Found at Index = 4


--------------------------------
Process exited after 2.2 seconds with return value 0
Press any key to continue . . .
```

THANKS