

# CAUC201 – Data Structures and Algorithms

## Assignment Submission

Assignment : 3

Student ID: 24BCA095

1.

```
#include <bits/stdc++.h>
using namespace std;
void bubble_sort(int a[],int n){
    bool swapped;
    for (int i = 0 ; i < n-1 ; i++){
        swapped = false;
        for (int j = 0 ; j < n-i-1 ; j++){
            if (a[j] > a[j+1]){
                swap(a[j], a[j+1]);
                swapped = true;
            }
        }
        if (!swapped) break;
    }
}
void linearsearch(int a[], int n)
{
    bubble_sort(a, n);
    cout << "Enter element to search : ";
    int key;
    cin >> key;
    for (int i = 0; i < n; i++)
    {
        if (a[i] == key)
        {
            cout << "Element is present " << endl;
            return;
        }
    }
    cout << "Element not found" << endl;
}

void binarysearch(int a[], int n)
{
    bubble_sort(a, n);
    int low, high, key;
    cout << "Enter element to search : ";
    cin >> key;
    low = 0;
    high = n - 1;
    while (low <= high)
    {
        int mid = (low + high) / 2;
        if (a[mid] == key)
```

## CAUC201 – Data Structures and Algorithms

```
{
    cout << "Element is present " << endl;
    return;
}
else if (a[mid] < key)
{
    low = mid + 1;
}
else
{
    high = mid - 1;
}
}
cout << "Element not found" << endl;
}
int main()
{
    int n;
    cout << "Enter size of array : ";
    cin >> n;
    int a[n];
    cout << "Enter " << n << " elements for the array: ";
    for (int i = 0; i < n; i++)
    {
        cin >> a[i];
    }
    cout << "Press 1 for linear search\nPress 2 for binary search\nChoice:";
    int ch;
    cin >> ch;
    switch (ch)
    {
        case 1:
            linearsearch(a, n);
            break;
        case 2:
            binarysearch(a, n);
            break;
        default:
            break;
    }
}
```

## CAUC201 – Data Structures and Algorithms

```
PS D:\JEET\COLLEGE\24BCA095\DSA\New folder\3>
Enter size of array : 5
Enter 5 elements for the array: 50
20
10
30
40
Press 1 for linear search
Press 2 for binary search
Choice:2
Enter element to search : 40
Element is present
```

2.

```
#include <iostream>
using namespace std;
int n;
void bubble_sort(char arr[])
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
    cout << "Sorted Data : ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}
void linear_search(char arr[], int n, char key)
{
    bubble_sort(arr);
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == key)
        {
            cout << "Element found at index " << i;
            return;
        }
    }
    cout << "Element wasn't found at any index";
}
void binary_search(char arr[], int n, char key)
{
```

## CAUC201 – Data Structures and Algorithms

```
bubble_sort(arr);
int left = 0, right = n - 1;
while (left <= right)
{
    int mid = (right + left) / 2;
    if (arr[mid] > key)
        right = mid - 1;
    else if (arr[mid] < key)
        left = mid + 1;
    else if (arr[mid] == key)
    {
        cout << "Data found at index " << mid;
        return;
    }
}
cout << "Data not found\n";
}

void size(int arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << "Enter value for index " << i << " : ";
        cin >> arr[i];
    }
}

int length(char arr[])
{
    int count = 0;
    while (arr[count] != '\0')
    {
        count++;
    }
    return count;
}

int main()
{
    int ch;
    char arr[100], key;
    cout << "Enter value in character : ";
    cin >> arr;
    n = length(arr);
    cout << "Enter value of key : ";
    cin >> key;
    cout << "Press 1 for Linear Search \nPress 2 for Binary Search\n";
    cin >> ch;
    switch (ch)
    {
        case 1:
            linear_search(arr, n, key);
```

## CAUC201 – Data Structures and Algorithms

	<pre>         break;     case 2:         binary_search(arr, n, key);         break;     } } </pre>
	<pre> PS D:\JEET\COLLEGE\24BCA095\DSA\New hing_char } Enter value in character : CMPICA Enter value of key : M Press 1 for Linear Search Press 2 for Binary Search 1 Sorted Data : A C C I M P Element found at index 4 </pre>
3.	<pre> #include &lt;iostream&gt; using namespace std; string key; int n; void bubble_sort(string arr[]); void linear_search(string arr[]) {     cout &lt;&lt; "Enter value of key : ";     cin &gt;&gt; key;     bubble_sort(arr);     for (int i = 0; i &lt; n; i++)     {         if (arr[i] == key)         {             //cout &lt;&lt; "String i.e. " &lt;&lt; arr[i] &lt;&lt; " was found at index " &lt;&lt; i;             cout &lt;&lt; "City is present" &lt;&lt; endl;             return;         }     }     cout &lt;&lt; "City is not present" &lt;&lt; endl; } void binary_search(string arr[]) {     cout &lt;&lt; "Enter value of key : ";     cin &gt;&gt; key;     bubble_sort(arr);     int left = 0, right = n - 1;     while (left &lt;= right)     {         int mid = (right + left) / 2;         if (arr[mid] &gt; key)             right = mid - 1;     } } </pre>

## CAUC201 – Data Structures and Algorithms

```
        else if (arr[mid] < key)
            left = mid + 1;
        else if (arr[mid] == key)
        {
            cout << "City is present" << endl;
            return;
        }
    }
    cout << "City is not present" << endl;
}

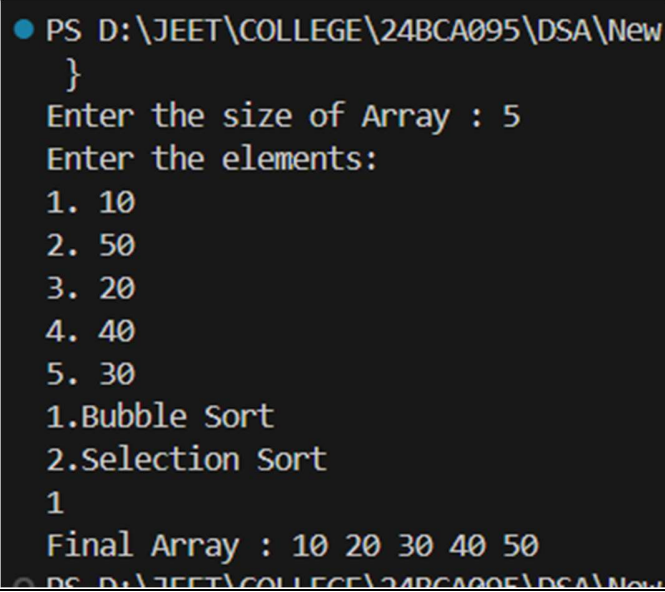
void bubble_sort(string arr[])
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
    cout << "Sorted Data : ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}

int main()
{
    cout << "Enter total number of cities : ";
    cin >> n;
    int ch;
    string arr[n];
    for (int i = 0; i < n; i++)
    {
        cout << "Enter name of city in CAPS for index " << i << " : ";
        cin >> arr[i];
    }
    cout << "Press 1 for linear Search\nPress 2 for binary search\nEnter choice : ";
    cin >> ch;
    switch (ch)
    {
        case 1:
            linear_search(arr);
            break;
        case 2:
            binary_search(arr);
            break;
        default:
            cout << "Invalid choice!!" << endl;
    }
}
```

## CAUC201 – Data Structures and Algorithms

	<pre> main(); break; } } </pre>
	<pre> ● PS D:\JEET\COLLEGE\24BCA095\DSA\New folder\3&gt; cd "d search } Enter total number of cities : 4 Enter name of city in CAPS for index 0 : MUMBAI Enter name of city in CAPS for index 1 : PUNE Enter name of city in CAPS for index 2 : AHMEDABAD Enter name of city in CAPS for index 3 : BANGALORE Press 1 for linear Search Press 2 for binary search Enter choice : 2 Enter value of key : PUNE Sorted Data : AHMEDABAD BANGALORE MUMBAI PUNE City is present </pre>
4.	<pre> #include &lt;bits/stdc++.h&gt; using namespace std; void bubble_sort(int arr[],int n){     bool swapped;     for(int i=0;i&lt;n-1;i++){         swapped = false;         for(int j=0;j&lt;n-i-1;j++){             if(arr[j]&gt;arr[j+1]){                 swap(arr[j],arr[j+1]);                 swapped = true;             }         }         if(!swapped) break;     } }  void selection_sort(int arr[],int n){     for(int i=0;i&lt;n-1;i++){         int minIndex=i;         for(int j=i+1;j&lt;n;j++){             if(arr[j]&lt;arr[minIndex])                 minIndex=j;         }         swap(arr[i],arr[minIndex]);     } }  void assign_value(int arr[],int n){     cout &lt;&lt; "Enter the elements: "&lt;&lt;endl;     for(int i=0;i&lt;n;i++){         cout &lt;&lt; (i+1) &lt;&lt; ". "; </pre>

## CAUC201 – Data Structures and Algorithms

	<pre>         cin&gt;&gt;arr[i];     }  } void display(int arr[], int n){     cout &lt;&lt; "Final Array : ";     for (int i = 0; i &lt; n; i++)         cout &lt;&lt; arr[i] &lt;&lt; " ";     cout &lt;&lt; endl; } int main(){     int n,ch;     cout &lt;&lt; "Enter the size of Array : ";     cin &gt;&gt; n;     int arr[n];     assign_value(arr, n);     cout &lt;&lt; "1.Bubble Sort" &lt;&lt; endl&lt;&lt;"2.Selection Sort" &lt;&lt; endl;     cin &gt;&gt; ch;     if(ch==1)         bubble_sort(arr, n);     else if(ch==2)         selection_sort(arr, n);     display(arr, n);     return 0; } </pre>
	 <pre> ● PS D:\JEET\COLLEGE\24BCA095\DSA\New } Enter the size of Array : 5 Enter the elements: 1. 10 2. 50 3. 20 4. 40 5. 30 1.Bubble Sort 2.Selection Sort 1 Final Array : 10 20 30 40 50 ● PS D:\JEET\COLLEGE\24BCA095\DSA\New </pre>
5.	<pre> #include &lt;bits/stdc++.h&gt; using namespace std; int n; void bubble_sort(char arr[]) {     bool swapped;     for (int i = 0; i &lt; n - 1; i++) </pre>



## CAUC201 – Data Structures and Algorithms

```
{
    swapped = false;
    for (int j = 0; j < n - i - 1; j++)
    {
        if (arr[j] > arr[j + 1])
        {
            swap(arr[j], arr[j + 1]);
            swapped = true;
        }
    }
    if (!swapped)
        break;
}
}

void selection_search(char arr[])
{
    for (int i = 0; i < n - 1; i++)
    {
        int minIndex = i;
        for (int j = i; j < n; j++)
        {
            if (arr[j] < arr[minIndex])
                minIndex = j;
        }
        swap(arr[minIndex], arr[i]);
    }
}

void arraysize(char arr[])
{
    int i = 0;
    while (arr[i++] != '\0')
        n++;
}

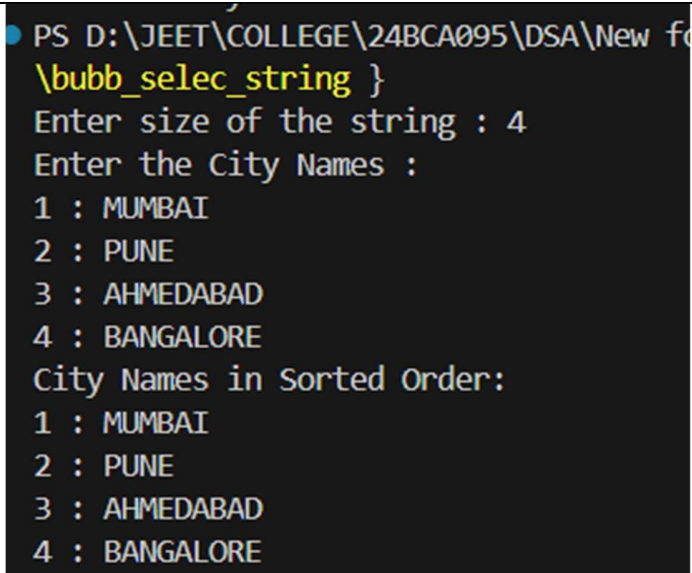
void display(char arr[])
{
    cout << "Sorted Array : ";
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
}

int main()
{
    int ch;
    char arr[100];
    cout << "Enter string : ";
    cin >> arr;
    arraysize(arr);
    cout << "1.Bubble Sort" << endl << "2.Selection Sort" << endl;
    cin >> ch;
    if (ch == 1)
        bubble_sort(arr);
```

## CAUC201 – Data Structures and Algorithms

	<pre> else     selection_search(arr); display(arr); } </pre>
	<pre> ● PS D:\JEET\COLLEGE\24BCA095\DSA\New folder } Enter string : CMPICA 1.Bubble Sort 2.Selection Sort 1 Sorted Array : A C C I M P </pre>
6.	<pre> #include &lt;iostream&gt; using namespace std; void bubble_sort(string arr[],int n){     bool swapped;     for (int i = 0 ; i&lt; n-1 ; i ++ ){         swapped = false;         for (int j = 0 ; j &lt; n - i - 1 ; j ++){             if(arr[j] &gt; arr[j+1]){                 swap(arr[j], arr[j+1]);                 swapped = true;             }         }         if (!swapped) break;     } } void selection_sort(string arr[],int n){     int minIndex;     for(int i = 0 ; i &lt; n - 1 ; i ++){         minIndex = i;         for(int j = i + 1 ; j &lt; n ; j ++){             if(arr[j] &lt; arr[minIndex]){                 minIndex = j;             }         }         swap(arr[i], arr[minIndex]);     } } void display(string arr[],int n){     cout &lt;&lt; "City Names in Sorted Order: \n";     for (int i = 0; i &lt; n; i++) {         cout &lt;&lt; (i+1) &lt;&lt; " : " &lt;&lt; arr[i] &lt;&lt; endl;     } } void assign(string arr[],int n){     cout &lt;&lt; "Enter the City Names : \n";     for (int i = 0 ; i&lt; n ; i++){ </pre>

## CAUC201 – Data Structures and Algorithms

	<pre>         cout &lt;&lt; (i+1) &lt;&lt; " : ";         cin &gt;&gt; arr[i];     } } int main(){     int n;     cout &lt;&lt; "Enter size of the string : ";     cin &gt;&gt; n;     string arr[n];     assign(arr, n);      display(arr,n); } </pre>
	 <pre> PS D:\JEET\COLLEGE\24BCA095\DSA\New fo \bubb_selec_string } Enter size of the string : 4 Enter the City Names : 1 : MUMBAI 2 : PUNE 3 : AHMEDABAD 4 : BANGALORE City Names in Sorted Order: 1 : MUMBAI 2 : PUNE 3 : AHMEDABAD 4 : BANGALORE </pre>
7.	<pre> #include &lt;bits/stdc++.h&gt; using namespace std; int size = 0; bool swapped; void assing_value(int arr[], int n); void bubble_sort(int arr[], int n) {     for (int i = 0; i &lt; n - 1; i++)     {         swapped = false;         for (int j = 0; j &lt; n - i - 1; j++)         {             if (arr[j] &gt; arr[j + 1])             {                 swap(arr[j], arr[j + 1]);                 swapped = true;             }         }     }     if (!swapped)         break; } </pre>

## CAUC201 – Data Structures and Algorithms

```
    }  
}  
void merge_sort(int arr1[], int arr2[], int arr[], int n1, int n2)  
{  
    bubble_sort(arr1, n1);  
    bubble_sort(arr2, n2);  
    int j = 0, i = 0;  
    while (i < n1 && j < n2)  
    {  
        if (arr1[i] < arr2[j])  
            arr[size++] = arr1[i++];  
        else if (arr2[j] < arr1[i])  
            arr[size++] = arr2[j++];  
        else if (arr1[i] == arr2[j])  
        {  
            arr[size++] = arr1[i++];  
            j++;  
        }  
    }  
    while (i < n1)  
        arr[size++] = arr1[i++];  
    while (j < n2)  
        arr[size++] = arr2[j++];  
}  
  
void assign_value(int arr[], int n)  
{  
    for (int i = 0; i < n; i++)  
        cin >> arr[i];  
}  
  
void display(int arr[])  
{  
    int i = 0;  
    cout << "Final Array : ";  
    while (i < size)  
        cout << arr[i++] << " ";  
}  
  
int main()  
{  
    int n1, n2;  
    cout << "Enter size of first array : ";  
    cin >> n1;  
    cout << "Enter size of second array : ";  
    cin >> n2;  
    int arr1[n1], arr2[n2], arr[n1 + n2];  
    cout << "Enter values for first array" << endl;  
    assign_value(arr1, n1);  
    cout << "Enter value for second array " << endl;  
    assign_value(arr2, n2);  
    merge_sort(arr1, arr2, arr, n1, n2);  
}
```

## CAUC201 – Data Structures and Algorithms

```
display(arr);  
return 0;  
}
```

```
PS D:\JEET\COLLEGE\24BCA095\DSA\New folder  
Enter size of first array : 4  
Enter size of second array : 5  
Enter values for first array  
1  
10  
2  
20  
Enter value for second array  
3  
30  
4  
40  
5  
Final Array : 1 2 3 4 5 10 20 30 40
```