**Assignment Submission**

**Assignment : 3**

**Student ID: 24BCA095**

|  |  |
| --- | --- |
|  | **#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)**  **{**  **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;**  **}**  **}** |
|  |  |
|  | **#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)**  **{**  **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);**  **break;**  **case 2:**  **binary\_search(arr, n, key);**  **break;**  **}**  **}** |
|  |  |
|  | **#include <iostream>**  **using namespace std;**  **string key;**  **int n;**  **void bubble\_sort(string arr[]);**  **void linear\_search(string arr[])**  **{**  **cout << "Enter value of key : ";**  **cin >> key;**  **bubble\_sort(arr);**  **for (int i = 0; i < n; i++)**  **{**  **if (arr[i] == key)**  **{**  **//cout << "String i.e. " << arr[i] << " was found at index " << i;**  **cout << "City is present" << endl;**  **return;**  **}**  **}**  **cout << "City is not present" << endl;**  **}**  **void binary\_search(string arr[])**  **{**  **cout << "Enter value of key : ";**  **cin >> key;**  **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 << "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;**  **main();**  **break;**  **}**  **}** |
|  |  |
|  | **#include <bits/stdc++.h>**  **using namespace std;**  **void bubble\_sort(int arr[],int n){**  **bool swapped;**  **for(int i=0;i<n-1;i++){**  **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\_sort(int arr[],int n){**  **for(int i=0;i<n-1;i++){**  **int minIndex=i;**  **for(int j=i+1;j<n;j++){**  **if(arr[j]<arr[minIndex])**  **minIndex=j;**  **}**  **swap(arr[i],arr[minIndex]);**  **}**  **}**  **void assign\_value(int arr[],int n){**  **cout << "Enter the elements: "<<endl;**  **for(int i=0;i<n;i++){**  **cout <<  (i+1) << ". ";**  **cin>>arr[i];**  **}**    **}**  **void display(int arr[], int n){**  **cout << "Final Array : ";**  **for (int i = 0; i < n; i++)**  **cout << arr[i] << " ";**  **cout << endl;**  **}**  **int main(){**  **int n,ch;**  **cout << "Enter the size of Array : ";**  **cin >> n;**  **int arr[n];**  **assign\_value(arr, n);**  **cout << "1.Bubble Sort" << endl<<"2.Selection Sort" << endl;**  **cin >> ch;**  **if(ch==1)**  **bubble\_sort(arr, n);**  **else if(ch==2)**  **selection\_sort(arr, n);**  **display(arr, n);**  **return 0;**  **}** |
|  |  |
|  | **#include <bits/stdc++.h>**  **using namespace std;**  **int n;**  **void bubble\_sort(char arr[])**  **{**  **bool swapped;**  **for (int i = 0; i < n - 1; i++)**  **{**  **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);**  **else**  **selection\_search(arr);**  **display(arr);**  **}** |
|  |  |
|  | #include <iostream>  using namespace std;  void bubble\_sort(string arr[],int n){      bool swapped;      for (int i = 0 ; i< n-1 ; i ++ ){          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\_sort(string arr[],int n){      int minIndex;      for(int i = 0 ; i < n - 1 ; i ++){          minIndex = i;          for(int j = i + 1 ; j < n ; j ++){              if(arr[j] < arr[minIndex]){                  minIndex = j;              }          }          swap(arr[i], arr[minIndex]);      }  }  void display(string arr[],int n){      cout << "City Names in Sorted Order: \n";      for (int i = 0; i < n; i++) {          cout << (i+1) << " : " << arr[i] << endl;      }  }  void assign(string arr[],int n){      cout << "Enter the City Names : \n";      for (int i = 0 ; i< n ; i++){          cout << (i+1) << " : ";          cin >> arr[i];      }  }  int main(){      int n;      cout << "Enter size of the string : ";      cin >> n;      string arr[n];      assign(arr, n);      display(arr,n);  } |
|  |  |
|  | #include <bits/stdc++.h>  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 < n - 1; i++)      {          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 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);      display(arr);      return 0;  } |
|  |  |