



Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information Technology

**Department of
Artificial Intelligence and Data Science**

Name: Siddhesh Dilip Khairnar

Class: SY

Division: B

Roll No: 272028

Semester: III

Academic Year: 2022-2023

Subject Name & Code: Data Structure, ADUA21202

Title of Assignment: Sort the data in ascending order using Selection sort (Display pass by pass output) and search a particular data using Binary search.

Assignment No.- 8

DS Assignment - 8

| |
|-----------|
| PAGE NO.: |
| DATE: / / |

Name : Siddhesh Dilip Khairnar

Division : B Roll no : 272028

PRN no : 272028

Aim : Sort the data in ascending order using selection sort & search using binary search.

Problem statement : Sort the data in ascending order using selection sort (display pass by pass output) and search a Particular data using binary search.

Background : →

Selection sort : The selection sort algorithm sort an array by repeatedly finding the minimum element from the unsorted part and putting it at the beginning. The algorithm maintain two subarray in a given array.

- The subarray which already sorted
- The remaining subarray was unsorted

Binary search : Binary search is an efficient algorithm for finding an item from a sorted list of item. It work by repeatedly dividing in half the portion of the list that could contain the item, until you've narrowed down the possible location to just one.

Software requirement : Online compiler or Any IDE.

Conclusion : Thus successfully learn how to sort the data in ascending order using selection sort which is display pass by pass output and search a Particular data using binary search.

©
Twin

Program:

```
VS Code > G+ selectionsort.cpp > ...
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int tot, arr[50], i, j, temp, small, chk, index;
6      cout<<"Enter the Size of Array: ";
7      cin>>tot;
8      cout<<"Enter "<<tot<<" Array Elements: ";
9      for(i=0; i<tot; i++)
10         cin>>arr[i];
11     for(i=0; i<(tot-1); i++)
12     {
13         chk=0;
14         small = arr[i];
15         for(j=(i+1); j<tot; j++)
16         {
17             if(small>arr[j])
18             {
19                 small = arr[j];
20                 chk++;
21                 index = j;
22             }
23         }
24         if(chk!=0)
25         {
26             temp = arr[i];
27             arr[i] = small;
28             arr[index] = temp;
29         }
30         cout<<"Step "<<i+1<<": ";
31     for(j=0; j<tot; j++)
32         cout<<arr[j]<<" ";
33     cout<<endl;
34     }
35 }
```

```

VS Code > binarysearch.cpp > ...
1 // Binary Search in C++
2
3 #include <iostream>
4 using namespace std;
5
6 int binarySearch(int array[], int x, int low, int high) {
7
8     // Repeat until the pointers low and high meet each other
9     while (low <= high) {
10         int mid = low + (high - low) / 2;
11
12         if (array[mid] == x)
13             return mid;
14
15         if (array[mid] < x)
16             low = mid + 1;
17
18         else
19             high = mid - 1;
20     }
21
22     return -1;
23 }
24
25 int main(void) {
26     int array[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
27     int x = 7;
28     int n = sizeof(array) / sizeof(array[0]);
29     int result = binarySearch(array, x, 0, n - 1);
30     if (result == -1)
31         printf("Not found");
32     else
33         printf("7 is found at index %d", result);
34 }

```

Output:

```

PS C:\Users\ABC\Downloads\VS Code> cd "c:\Users\ABC\Downloads\VS Code"
sort }
Enter the Size of Array: 10
Enter 10 Array Elements: 2 4 8 5 1 0 3 6 9 7
Step 1: 0 4 8 5 1 2 3 6 9 7
Step 2: 0 1 8 5 4 2 3 6 9 7
Step 3: 0 1 2 5 4 8 3 6 9 7
Step 4: 0 1 2 3 4 8 5 6 9 7
Step 5: 0 1 2 3 4 8 5 6 9 7
Step 6: 0 1 2 3 4 5 8 6 9 7
Step 7: 0 1 2 3 4 5 6 8 9 7
Step 8: 0 1 2 3 4 5 6 7 9 8
Step 9: 0 1 2 3 4 5 6 7 8 9

```

```

PS C:\Users\ABC\Downloads\VS Code> cd "c:\Users\ABC\Downloads\VS Code"
h }
7 is found at index 7

```