	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

PAGE NO.:
DATE: / /

DS Assignment - 8

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 begining. 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 effecient algorithm for finding an item from a sorted list of item. It work by repeatly 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.

Program:

VS Code > selectionsort.cpp > ...

```cpp
#include<iostream>
using namespace std;
int main()
{
    int tot, arr[50], i, j, temp, small, chk, index;
    cout<<"Enter the Size of Array: ";
    cin>>tot;
    cout<<"Enter "<<tot<<" Array Elements: ";
    for(i=0; i<tot; i++)
        cin>>arr[i];
    for(i=0; i<(tot-1); i++)
    {
        chk=0;
        small = arr[i];
        for(j=(i+1); j<tot; j++)
        {
            if(small>arr[j])
            {
                small = arr[j];
                chk++;
                index = j;
            }
        }
        if(chk!=0)
        {
            temp = arr[i];
            arr[i] = small;
            arr[index] = temp;
        }
        cout<<"Step "<<i+1<<": ";
for(j=0; j<tot; j++)
    cout<<arr[j]<<" ";
cout<<endl;
    }
}
```

VS Code > binarysearch.cpp > ...

```cpp
// Binary Search in C++

#include <iostream>
using namespace std;

int binarySearch(int array[], int x, int low, int high) {

    // Repeat until the pointers low and high meet each other
  while (low <= high) {
    int mid = low + (high - low) / 2;

    if (array[mid] == x)
      return mid;

    if (array[mid] < x)
      low = mid + 1;

    else
      high = mid - 1;
  }

  return -1;
}

int main(void) {
  int array[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
  int x = 7;
  int n = sizeof(array) / sizeof(array[0]);
  int result = binarySearch(array, x, 0, n - 1);
  if (result == -1)
    printf("Not found");
  else
    printf("7 is found at index %d", result);
}
```

Output:

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