Activity No. <n>	
<Replace with Title>	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: 9/11/25
Section: CPE21S4	Date Submitted: 9/11/25
Name(s): Alcantara, Jason P.	Instructor: Engr. Jimlord Quejado

6. Output:

1. **What is a search tree in data structures?**
   - **This is a type of Binary tree where each of the nodes contains some specials key and fulfill a specific type of ordering properties. This search trees are widely used because they gave a good balance offer of structured searching.**

2. **What are the Different types of search algorithm in data structures? Differentiate each type of search.**
   - **There are types of algorithms in data structure has distinct characteristics and applications, And some has a different conditions and how settle the algorithms. Some type is linear search where we can check each elements from the start till the end until it identifies its target elements. The difference between the two is that it does not require the data to be sorted since its low for some large data based. For Binary search, its algorithm divides the search interval in half where it compares the target value with the middle element of the sorted data. This type requires the data to be sorted not like linear search and it is more efficient than linear search.**

3. **What operations / implementations can be performed using binary and linear search operations?**
   - **For Linear search, it's implementations is that, it can support the sequential traversal like arrays and linked list. For Binary search, it implemented the sorted arrays. Linear search can easily modify to find its minimum or maximum element in some sorted list, while binary search categorize the list at the minimum element where it starts at the beginning and for maximum is at the end.**

4. **What are the advantages in using binary search tree as data structure?**
   - **The advantages while using binary search is that it can sort out data, making it easier to retrieve data in a specific order. Adding and removing elements in binary search is much more simpler and efficient since it looks like its taking a shortcut. One advantages is that binary is much more quicker than linear because of the data that needs to be searched half of the elements in each steps then locating the element at the center dividing it to 2.**

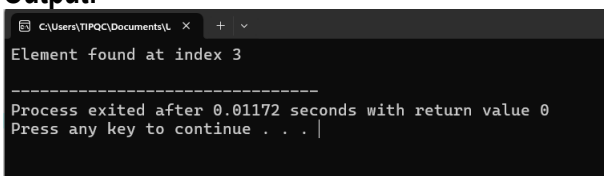
5. **Give an example program using binary search and Linear search.**

```
#include <iostream>
using namespace std;

// Function to perform Linear Search
int linearSearch(int arr[], int size, int target) {
   for(int i = 0; i < size; i++) {
      if(arr[i] == target)
         return i; // Target found at index i
   }
   return -1; // Target not found
}
```

```
int main() {
   int data[] = {34, 78, 12, 56, 89, 23};
   int size = sizeof(data)/sizeof(data[0]);
   int target = 56;
   int result = linearSearch(data, size, target);
   if(result != -1)
      cout << "Element found at index " << result << endl;
   else
      cout << "Element not found in the array." << endl;
   return 0;
}
```

Output:



7. Supplementary Activity

8. Conclusion:

9. Assessment Rubric