ASSIGNMENT-1

1. Make a single linked list of integers. There should be at least 15 nodes. The list should not be sorted.

Traverse the list.

Now sort the list using Bubble sort. /do not use any other sorting algorithm. The list should be sorted such that your program unlinks the nodes and relinks them so that they are sorted. (DO NOT SWAP THE VALUES IN THE NODES).

use Bubble sort.

Traverse the list again.

Submit the complete code.

A readme file with instructions to compile.

submit each screen shot of your program execution.

Code :

```
#include<iostream>
using namespace std;
class Node
{
	public:
	int data;
	Node* next;
};
// Adding a node in the linked list
void AddNode(Node** head, int a)
{
	Node* nextNode = new Node();
	Node* curr = *head;
	nextNode->data = a;
	nextNode->next = NULL;
	if (curr == NULL)
	{
		*head = nextNode;
		return;
	}
	while (curr->next != NULL)
	{
		curr = curr->next;
	}
	curr->next = nextNode;
	return;
}
//Creating LinkedList
void create(Node** head)
{
	int n, a;
	cout << "Enter the number of nodes to be inserted" << endl;
	cin >> n;
	for (int i = 0; i < n; i++)
	{
		cout << "Enter the value of the node" << endl;
		cin >> a;
		AddNode(head, a);
	}
	return;
}
//Printing the LinkedList
void print(Node* head)
{
	Node* curr = head;
	while (curr != NULL)
	{
		cout << curr->data << " ";
		curr = curr->next;
	}
	cout << endl;
	return;
}
//BubbleSort with swapping values
void BubbleSort(Node** head)
{
	//Node* curr = *head;
	Node* curr, *curr2;
	bool swapped = true;
	while (swapped == true)
	{
		swapped = false;
		curr = *head;
		curr2 = curr->next;
		while (curr2 != NULL)
		{
			if (curr->data > curr2->data)
			{
				int tmp = curr->data;
				curr->data = curr2->data;
				curr2->data = tmp;
				swapped = true;
			}
			curr = curr2;
			curr2 = curr2->next;
		}
	}
	return;
}
//BubbleSort by swapping Nodes
void BubbleSortSwapNodes(Node** head)
{
	Node* curr, * curr2, * prev;
	bool swapped = true;
	while (swapped == true)
	{
		swapped = false;
		prev = NULL;
		curr = *head;
		curr2 = curr->next;
		while (curr2 != NULL)
		{
			if (curr->data > curr2->data)
			{
				if (prev == NULL)
				{
					*head = curr2;
				}
				else
				{
					prev->next = curr2;
				}
				Node* n2 = curr2->next;
				curr2->next = curr;
				curr->next = n2;
				swapped = true;
			}
			prev = curr;
			curr = curr2;
			curr2 = curr2->next;
		}
	}
	return;
}
int main()
{
	Node* head = NULL;
	create(&head);
	cout << "Original unsorted List is " << endl;
	print(head);
	//BubbleSortSwappingNodes
	BubbleSortSwapNodes(&head);
	cout << "Sorted List after swapping nodes is " << endl;
	print(head);
	return 0;
}
```

Execution Steps :

1) Build the code file.

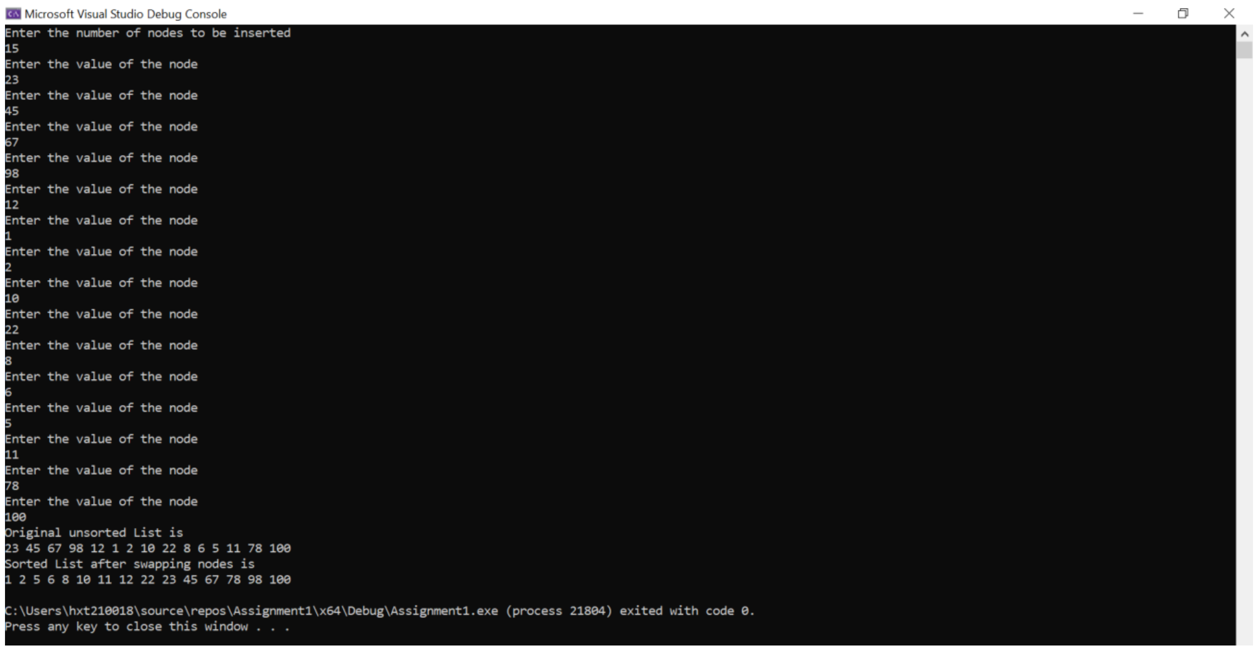
2) Run the program. -> Enter the number of the nodes to be inserted.

-> Enter the value of the nodes.

-> The program will print the given unsorted list

-> The program runs the bubble sort and print the sorted list.

Screenshot of execution :



```
Enter the number of nodes to be inserted
15
Enter the value of the node
23
Enter the value of the node
45
Enter the value of the node
67
Enter the value of the node
98
Enter the value of the node
12
Enter the value of the node
1
Enter the value of the node
2
Enter the value of the node
10
Enter the value of the node
22
Enter the value of the node
8
Enter the value of the node
6
Enter the value of the node
5
Enter the value of the node
11
Enter the value of the node
78
Enter the value of the node
100
Original unsorted List is
23 45 67 98 12 1 2 10 22 8 6 5 11 78 100
Sorted List after swapping nodes is
1 2 5 6 8 10 11 12 22 23 45 67 78 98 100

C:\Users\hxt210018\source\repos\Assignment1\x64\Debug\Assignment1.exe (process 21804) exited with code 0.
Press any key to close this window . . .
```

2. We covered binary search algorithm for an array. Write program similar to binary search, but now divide the list into 3 parts each time. So this would be tertiary search algorithm.

Your program must be recursive,

run it 2 times .. once try to search a number in the list.

Second time search a number not in the list.

Submit the code and screen shot of executions.

Code :

```
#include<iostream>
using namespace std;
int TernarySearchRecursive(int A[], int low, int high, int v);
int main()
{
	int n, A[1000], v;
	cout << "Enter the number of elements" << endl;
	cin >> n;
	cout << " Enter the array values in sorted order" << endl;
	for (int i = 0; i < n; i++)
	{
		cin >> A[i];
	}
	bool ans;
	do
	{
		cout << "Enter value to be searched" << endl;
		cin >> v;
		int loc = TernarySearchRecursive(A, 0, n - 1, v);
		if (loc == -1)
			cout << " ELement not present in the array" << endl;
		else
			cout << "Found at " << loc << " location in the array" << endl;
		cout << "Is there any other value to be searched[Enter 0(no) or 
1(yes)]?" << endl;
		cin >> ans;
	} while (ans == true);
	return 0;
}
int TernarySearchRecursive(int A[], int low, int high, int v)
{
	if (low > high)
		return -1;
	int mid1 = low + (high - low) / 3;
	int mid2 = high - (high - low) / 3;
	if (A[mid1] == v) return mid1;
	if (A[mid2] == v) return mid2;
	if (A[mid1] > v)
		return TernarySearchRecursive(A, low, mid1 - 1, v);
	else
	if (A[mid1] < v && A[mid2] > v)
		return TernarySearchRecursive(A, mid1 + 1, mid2 - 1, v);
	else
		return TernarySearchRecursive(A, mid2 + 1, high, v);
}
```

Execution steps :

1) Build the code file.

2) Run the program. -> (i) Enter the number of elements in the array[Maximum number given is 1000].

-> (ii) Enter the array elements in sorted order.

-> (iii) Enter the value to be searched.

-> (iv) The program runs the ternary search and print the location of the value if present in the array . Else it prints “Element not present in the array”.

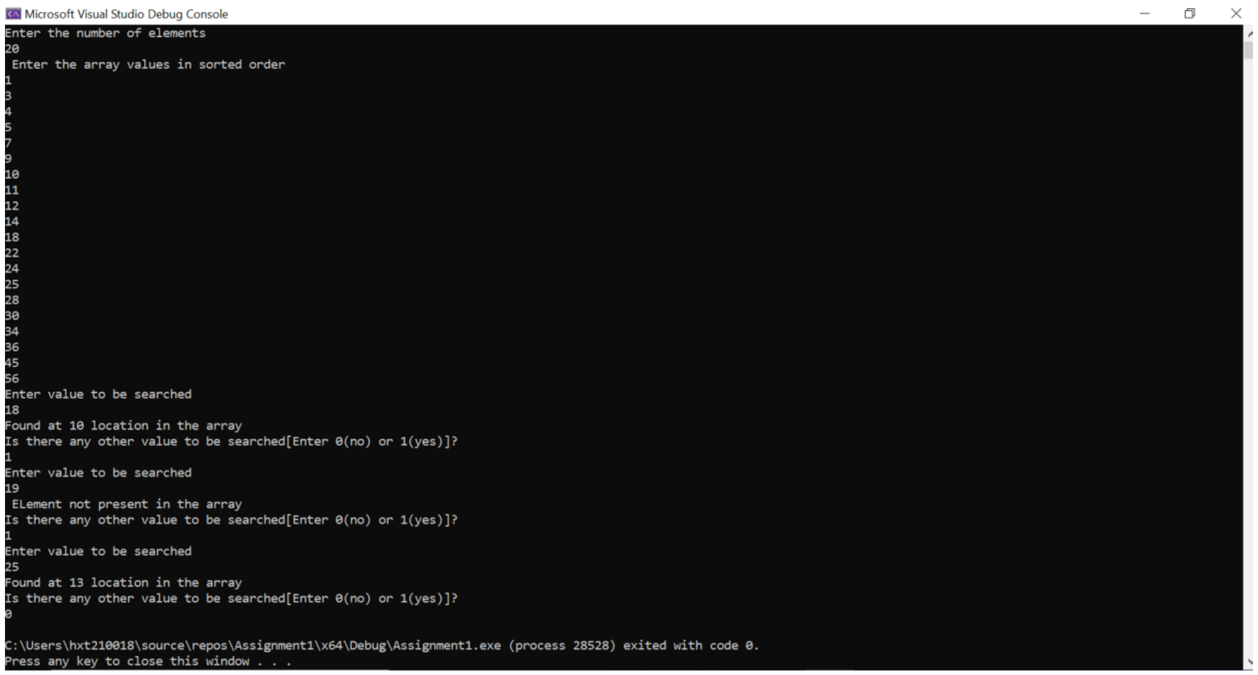
-> (v) It will ask for a prompt whether you want to search any other value.

-> (vi) Enter 1 for yes or 0 for no.

-> (vii) If yes, it will repeat the steps (iii) to (v).

-> (viii) If no, the program terminates.

Screenshot of code execution :



Microsoft Visual Studio Debug Console

```
Enter the number of elements
20
 Enter the array values in sorted order
1
3
4
5
7
9
10
11
12
14
18
22
24
25
28
30
34
36
45
56
Enter value to be searched
18
Found at 10 location in the array
Is there any other value to be searched[Enter 0(no) or 1(yes)]?
1
Enter value to be searched
19
 ELement not present in the array
Is there any other value to be searched[Enter 0(no) or 1(yes)]?
1
Enter value to be searched
25
Found at 13 location in the array
Is there any other value to be searched[Enter 0(no) or 1(yes)]?
0

C:\Users\hxt210018\source\repos\Assignment1\x64\Debug\Assignment1.exe (process 28528) exited with code 0.
Press any key to close this window . . .
```