01-08-2024

SINGLY LINKED LIST & SORT IT USING INSERTION SORT ALGORITHM =>

```
#include <stdio.h>
#include <stdlib.h>
struct Node
{
    int data;                    // data = info
    struct Node* next;
};
struct Node* createNode(int data)
{
    struct Node* newNode = (struct Node*) malloc (sizeof(struct Node));
    newNode -> data = data;
    newNode -> next = NULL;
    return newNode;
}
void insertEnd(struct Node** head, int data)
{
    struct Node* newNode = createNode(data);
    if (*head == NULL)
    {
        *head = newNode;
        return;
    }
    // Traverse to the end of the list
    struct Node* temp = *head;
    while (temp -> next != NULL)
    {
        temp -> next = newNode;
        temp = temp -> next;
    }
```

```
    temp -> next = newNode;
}

// To print Linked list
void printList(struct Node* head)
{
    while (head != NULL)
    {
        printf("%d ", head -> data);
        head = head -> next;
    }
}

// To sort Linked list using Insertion Sort
void insertionSort(struct Node** head)
{
    struct Node* sorted = NULL;          // sorted list is EMPTY
    struct Node* current = *head;        // Traverse the given Linked list
                                         & insert every node to
                                         sorted
    while (current != NULL)
    {
        struct Node* next = current -> next;
        if (sorted == NULL || sorted -> data >= current -> data)
        {
            current -> next = sorted;
            sorted = current;
        }
        else
        {
            struct Node* temp = sorted;  // Locate node before the
                                         point of insertion.
            while (temp -> next != NULL && temp -> next ->
data < current -> data)
```

```
    {
        temp = temp -> next;
    }
    current -> next = temp -> next;
    temp -> next = current;
    }
    current = next;
}
*head = sorted;
}
int main()
{
    struct Node* head = NULL;    // Initialize head as NULL
    int n, data, i;              // data = info
    printf("Enter no of elements to be sorted: ");
    scanf("%d", &n);
    for (i=0; i<n; i++)          // Insert elements into Linked List
    {
        printf("Enter element %d: ", i+1);
        scanf("%d", &data);
        insertEnd(&head, data);
    }
    printf("\n original List: ");
    printList(head);             // Print's original list
    insertionSort(&head);        // Sort the Linked list
    printf("\n Sorted List: ");
    printList(head);             // Print the sorted list
}
```