* Sort using single linked list:

=>

```
#include <stdio.h>
#include <stdlib.h>

struct Node{
        int data;
        struct Node * next;
};

void display (struct Node * head){
        struct Node * current = head;
        while (current != NULL){
                printf("%d H", current->dd
                current = current -> next;
        }
        printf("NULL\n");
}

struct Node * Sort LinkedList (struct Node*
 head){
        if (head == NULL || head -> next == Nul)
                return head;
        }
```

Only Sorting completed

29/11/24

```
int swapped;
struct Node *temp;
struct Node *end = NULL;

do {
    swapped = 0;
    struct Node* current = head;

    while (current -> next != en
        if (current -> data > curre
        -> next -> data) {

            int tempData = curren
            data;
            current -> data = curren
            next -> data;
            current -> next -> data =
            temp Data;

            swapped = 1;
        }
        current = current -> nex
    }
    end = current;

} while (swapped);
return head;
}
```

```
int main(){

    struct Node * node1 = (struct Node *)
    malloc (sizeof (struct Node));

    struct Node * node2 = (struct Node *)
    malloc (size of (struct Node));

    struct Node * node3 = (struct Node *)
    malloc (size of (struct Node));

    struct Node * node4 = (struct Node *
    malloc (size of (struct Node));

    node1 -> data = 4;
    node2 -> data = 2;
    node 3 -> data = 7;
    node 4 -> data = 1;

    node1 -> next = node 2;
    node 2 -> next = node 3;
    node 3 -> next = node 4;
    node 4 -> next = NULL;

    ~~printf ("sorted~~
```

```
printf ("Original linked list \n");
display (node1);

node1 = sortlinkedList (node 1);

printf ("Sorted linked list: \n");
display (node 1);

free (node 1)
free (node 2)
free (node 3)
free (node 4)

return 0;

}
```

```
Original Linked List:
4    2    7    1    NULL
Sorted Linked List:
1    2    4    7    NULL
```