

○ WAY to sort / concatenate list, reverse a list

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

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

```
struct node {
```

```
    int data;
```

```
    struct Node *next;
```

```
}
```

```
struct node * createNode (int data) {
```

```
    struct node * newnode = (struct node *) malloc
```

(size of struct

Node);

```
    newnode->data = data;
```

```
    newnode->next = NULL;
```

```
    return newnode;
```

```
}
```

```
void InsertAtEnd (struct node * head, int data)
```

```
{
```

```
    struct Node * newnode = createNode (data);
```

```
    if (head == NULL) {
```

```
        head = newnode;
```

```
}
```

```
    struct Node * temp = head;
```

```
    while (temp->next != NULL) {
```

```
        temp = temp->next
```

```
}
```

```
    temp->next = newnode
```

```
}
```

```
void displaylist ( struct node * head ) {
```

```
while ( head != NULL ) {
```

```
printf ( "%d", head->data );
```

```
head = head->next;
```

```
}
```

```
}
```

```
void sortlist ( struct node * head ) {
```

```
int temp;
```

```
struct Node * ptr2;
```

```
struct node * ptr2 = NULL;
```

```
if ( head == NULL ) {
```

```
printf ( "Empty list" );
```

```
}
```

```
for ( ptr2 = head; ptr2->next != NULL; ptr2 = ptr2->next )
```

```
{
```

```
for ( ptr2 = ptr2->next; ptr2 != NULL; ptr2 = ptr2->next )
```

```
{
```

```
if ( ptr1->data > ptr2->data ) {
```

```
temp = ptr1->data;
```

```
ptr1->data = ptr2->data;
```

```
ptr2->data = temp;
```

```
}
```

```
}
```

```
}
```

```
}
```



```

void reverseList(struct Node **head) {
    struct Node *prev = NULL;
    struct Node *current = *head;
    struct Node *next;

    while (current != NULL) {
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    *head = prev;
}

```

```

void concatenateList(struct Node **list1, struct Node **list2) {

```

```

    if (*list1 == NULL) {

```

```

        *list1 = *list2;

```

```

    }

```

```

    else {

```

```

        struct Node *temp = *list1;

```

```

        while (temp->next != NULL) {

```

```

            temp = temp->next;

```

```

            temp->next = *list2;

```

```

        }

```

```

    }

```

int main() {

struct node * list1 = NULL;

struct node * list2 = NULL;

insertatend(&list2, 5);

insertatend(&list1, 3);

insertatend(&list1, 8);

insertatend(&list1, 12);

insertatend(&list2, 10);

insertatend(&list2, 7);

printf("Original list 1: ");

displaylist(list1);

printf("Original list 2: ");

displaylist(list2);

Sortlist(list1);

printf("Sorted list 1: ");

displaylist(list1);

Reverse list(&head2);

concatenatedlist(&list1, list2);

printf("Concatenated list: ");

displaylist(list1);

printf("Shreyas Athiga IBM ZRC 5269");

return 0;

Output: Original list 1: 3 → 5 → 8 → NULL

Original list 2: 12 → 10 → 7 → NULL

Sorted list: 3 → 5 → 8 → NULL

Reverse list: 7 → 10 → 12 → NULL

Concatenated list: 3 → 5 → 8 → 7 → 10 → 12 → NULL;

Shreyas Athiga IBM ZRC 5269

See
26.02.21