

7-12-2020

AVIRATH MANOHAR JECRC
IBMI9CS195

LAB 7 & LAB 8.

→ Sort-1

struct node *ptr = *head

struct node *temp = NULL

int i

if (head == NULL) return

else {

while (ptr != NULL) {

temp = ptr → next

while (temp != NULL) {

if (ptr → data > temp → data) {

i = ptr → data

ptr → data = temp → data

temp → data = i

}

temp = temp → next;

}

ptr = ptr → next;

}

}

}

reverse() {

struct node *prev = NULL

struct node *next = NULL

struct node *ptr = *head

while (ptr != NULL) {

next = ptr → next

ptr → next = prev

prev = ptr

ptr = next

}

```
*head = prev
```

```
}
```

```
concatenate (struct node *ptu1, struct node *ptu2)
```

```
{
```

```
if (ptu1 != NULL && ptu2 == NULL)
```

```
{
```

```
if (ptu1->next == NULL)
```

```
ptu1->next = ptu2
```

```
else
```

```
concatenate (ptu1->next, ptu2)
```

```
}
```

```
else { printf: "Either ptu1/ptu2 is NULL" }
```

```
}
```

```
struct node *concat (struct node *start1, struct node *start2)
```

```
{
```

```
struct node *ptr
```

```
if (start1 == NULL)
```

```
{
```

```
start1 = start2
```

```
return start1
```

```
}
```

```
if (start2 == NULL)
```

```
return start1
```

```
ptr = start1
```

```
while (ptr->next != NULL)
```

```
ptr = ptr->next
```

```
ptr->next = start2
```

```
return start1
```

```
}
```


Pop () {

struct node * ptr

if (head == NULL)

printf ("List is Empty")

else {

ptr = head

head = ptr -> next

free (ptr)

}

}

Enqueue (item) {

struct node * ptr & temp

ptr = (struct node *) malloc (size of (struct node))

ptr -> data = item

ptr -> next = NULL

if (head == NULL)

{

head = ptr

}

else

{

temp = head

while (temp -> next != NULL)

{

temp = temp -> next

}

temp -> next = ptr

}

}

1) dequeue() {

struct node *ptr;

if (head == NULL)

{

Print " list is empty "

}

else

{

ptr = head

head = ptr -> next

free(ptr)

}

}