

SINGLY LINKED LISTS IMPLEMENTATION

No.

```
struct Node  
{
```

```
    int data
```

```
    struct Node * next  
}
```

```
push-front (struct Node * head, int new-data)  
{
```

```
    struct Node * new-node = (struct Node *) malloc (sizeof  
        (struct Node)).
```

```
    new-node -> data = new-data
```

```
    new-node -> next = *head
```

```
    *head = new-node  
}
```

```
push-end (struct Node ** head, int new data)  
{
```

```
    struct Node * new node = (struct Node *) malloc  
        (sizeof (struct Node))
```

```
    struct Node * last = *head
```

```
    new-node -> next = NULL
```

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

```
        *head = new-node
```

```
        return;  
    }
```

```
}
```

MESID
RAZZ

```

while (last → next != NULL)
    last = last → next
last → next = new_node

```

```

return;
}

```

```

push_specific_pos (int data, int position)
{

```

```

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

```

```

    new_node → data = data;

```

```

    int i;

```

```

    struct Node * temp = head

```

```

    if (position == 1)
    {

```

```

        new_node → next = temp

```

```

        head = new_node

```

```

        return
    }

```

```

    for (i = 1; i < position - 1; i++)
    {

```

```

        1

```

```

        temp = temp → next

```

```

        new_node → next = temp → next

```

```

        temp → next = new_node
    }

```

MESID
RIZZ

print_Linkedlist (struct Node * node)

{

while (node != NULL)

{

printf("%d", node->data)

node = node->next

}

}

delete_front()

{

struct Node * ptr;
if (head == NULL)

{

printf("List is Empty")

}

else

{

ptr = head

head = ptr->next

free(ptr)

}

}

delete_end()

{

MESID
RAZZ

```

Struct Node * ptr & ptr1;
if (head == NULL)
    // list is empty
else if (head->next == NULL)
{
    head = NULL;
    free(head);
}
else
{
    ptr = head;
    while (ptr->next != NULL)
    {
        ptr1 = ptr;
        ptr = ptr->next;
    }
    ptr->next = NULL;
    free(ptr);
}

```

delete specific pos()

```

Struct Node * ptr, * ptr1;
int i, position

```

MESID
RAZZ

scanf ("%d", &position)

ptr = head

for (i = 0; i < position; i++)

{

ptr = ptr->next

ptr = ptr->next

if (ptr == NULL)

{

less than required element in the list

return;

}

}

ptr->next = ptr->next

free(ptr)

}

MESID
RIZZ