

## Doubly linked list

struct

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

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

```
struct Node
```

```
int data;
```

```
struct Node *left;
```

```
struct Node *right;
```

```
};
```

```
typedef struct Node node;
```

```
node *start = NULL;
```

```
node *new1, *curr, *ptr;
```

```
void create();
```

```
void display();
```

```
void insertleft();
```

```
void DeleteSpecificElement();
```

```
void main() {
```

```
int ch;
```

```
while (1) {
```

```
printf("\n 1. Create 2. Display 3. Insert left 4. Delete
```

```
Specific Element 5. Exit");
```

```
printf("\n Enter Your choice: ");
```

```
scanf("%d", &ch);
```

```
switch (ch) {
```

```
case 1: create();
```

```
break;
```

```
case 2: display();
```

```
break;
```

case 3: InsertLeft();

break;

case 4: DeleteSpecificElement();

break;

case 5: exit(0);

}

}

}

void create() {

char ch;

do {

new1 = (node \*) malloc (sizeof (node));

printf("Enter Value : ");

scanf("%d", &new1->data);

new1->left = NULL;

new1->right = NULL;

if (start == NULL) {

start = new1;

~~start = new1;~~

curr = new1;

} else {

curr->right = new1;

new1->left = curr;

curr = new1;

}

printf("Do You Want to Add an Element (Y/N)? ");

scanf("%c", &ch);

while (ch == 'Y' || ch == 'y');

}

```
void display () {
```

```
    if (start == NULL) {
```

```
        printf("\n Linked list is Empty ");
        return;
    }

```

```
    ptr = start;
```

```
    printf("\n Elements in linked list: \n");
```

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

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

```
        ptr = ptr->right;
    }

```

```
}

```

```
void InsertLeft () {
```

```
    int val;
```

```
    printf("Enter Value: ");
```

```
    scanf("%d", &val);
```

```
    new 1 = (node *) malloc (sizeof (node));
```

```
    new 1->data = val;
```

```
    new 1->left = NULL;
```

```
    new 1->right = NULL;
```

```
    printf("Enter Value to Insert Left of: ");
```

```
    scanf("%d", &val);
```

```
    ptr = start;
```

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

```
        new 1->right = ptr;
```

```
        new 1->left = ptr->left;
```

```
        if (ptr->left != NULL) {
```

```
            ptr->left->right = new 1;
        }
    }

```

```


```

```
    ptr->left = new 1;
```

```
    if (ptr == start) {
```

```
start = new 1;
```

```
}
```

```
{ else {
```

```
printf("Value Not found.");
```

```
}
```

```
}
```

```
void DeleteSpecificElement()
```

```
{ int value;
```

```
printf("In Enter Value to delete: ");
```

```
scanf("%d", &value);
```

```
ptr = start;
```

```
while(ptr != NULL && ptr->data != value){
```

```
ptr = ptr->right;
```

```
}
```

```
if (ptr == NULL){
```

```
printf("Value Not found");
```

```
return;
```

```
}
```

```
if (ptr->left != NULL){
```

```
ptr->left->right = ptr->right;
```

```
}
```

```
if (ptr->right != NULL){
```

```
ptr->right->left = ptr->left;
```

```
}
```

```
if (ptr == start){
```

```
start = ptr->right;
```

```
}
```

```
free(ptr);
```

```
printf("Element with value %d deleted", value);
```

```
}
```

Output

1. Create

2. Display

3. Insert Left

4. Delete Specific Element

5. Exit

→ Enter Your Choice : 1

Enter Value : 10

Do You Want to Add an Element (Y/N)? y

Enter Value : 20

Do You Want to Add an Element (Y/N)? y

Enter Value : 30

Do You Want to Add an Element (Y/N)? n

→ Enter Your Choice : 2

Elements in Linked List:

10 20 30

→ Enter Your Choice : 3

Enter Value : 40

Enter the Value to Insert Left of : 20

→ Enter Your Choice : 2

Elements in Linked List:

10 40 20 30

→ Enter Your Choice : 4

Enter Value to Delete : 20

Element with value 20 deleted

~~Box~~

→ Enter Your Choice: 2

Elements in Linked List:

10 40 30

→ Enter Your Choice: 5



3/12/2024