

/*WAP to Implement Singly Linked List with following operations a) Create a linked list. b) Deletion of first element, specified element and last element in the list. c) Display the contents of the linked list. */

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

struct node
{
    int info;
    struct node * next;
};

struct node* createsll()
{
    struct node *p;
    struct node *start = NULL, *follow = NULL;
    int item;
    printf("Enter the element (write -999 to exit)\n");
    scanf("%d", &item);
    while (item != -999)
    {
        p = (struct node*)malloc(sizeof(struct node));
        p->info = item;
        p->next = NULL;
        if (start == NULL)
        {
            start = p;
        }
        else
        {
            follow->next = p;
            follow = p;
        }
    }
}
```

```
    follow = p;

}

else

{

    follow->next = p;

    follow = p;

}

scanf("%d", &item);

}

return start;

}

struct node * deletefirst(struct node * start)

{

    struct node *temp;

    if(start==NULL)

    {

        printf("The linked list is empty");

    }

    else if (start->next ==NULL)

    {

        printf("The deleted element is %d",start->info);

        temp=start;

        start=NULL;

        free(temp);

    }

}
```

```
    }

else

{

    temp=start;

    start=start->next;

    printf("The deleted element is %d",temp->info);

    free(temp);

}

return start;

}

struct node * deletelast(struct node * start)
```

```
{

struct node *temp,*follow;

if(start==NULL)

{

    printf("The linked list is empty");

}

else if (start->next ==NULL)

{

    printf("The deleted is %d",start->info);

    temp=start;

    start=NULL;

    free(temp);
```

```
    }

else

{
    temp=start;

    while(temp->next!=NULL)

    {

        follow=temp;

        temp=temp->next;

    }

    follow->next=NULL;

    printf("The deleted is %d",temp->info);

    free(temp);

}

return start;

}
```

```
struct node * delete(struct node * start,int item)

{

    struct node *temp,*follow;

    if(start==NULL)

    {

        printf("The linked list is empty");

    }

    else if (start->next ==NULL)

    {
```

```
printf("The deleted is %d",start->info);

temp=start;

start=NULL;

free(temp);

}

else

{



temp=start;

while( temp->next!=NULL && temp->info != item)

{

follow=temp;

temp=temp->next;

}

printf("The deleted is %d",temp->info);

follow->next=temp->next;

free(temp);

}

return start;

}

void dis(struct node * start)

{

struct node * temp;

if(start==NULL)
```

```
{  
    printf("The linked list is empty");  
}  
  
else  
{  
    temp=start;  
    printf("the elements are:-\n");  
    while(temp!=NULL)  
    {  
        printf("%d \n",temp->info);  
        temp=temp->next;  
    }  
}  
  
}  
  
int main()  
{  
    struct node * head=NULL;  
    int value;  
    int choice;  
    while(1)  
    {  
        printf("\n 1.Create linked list \n");  
        printf(" 2.delete at front \n 3.delete at end\n 4.delete an element \n"  
5.disaplay \n 6.exit\n");
```

```
printf("enter the choice \n");
scanf("%d",&choice);
switch(choice)
{
    case 1: head=createsll();
              break;
    case 2: head=deletefirst(head);
              break;
    case 3:head=deletelast(head);
              break;
    case 4:printf("Enter the value to be deleted: ");
              scanf("%d",&value);
              head=delete(head,value);
              break;
    case 5:dis(head);
              break;
    case 6:printf("Exiting program");
              return 0;
    default:return 0;
}
}
```

OUTPUT:-

PS C:\Users\88nin\OneDrive\Documents\Data structure> cd 'c:\Users\88nin\OneDrive\Documents\Data structure\output'
PS C:\Users\88nin\OneDrive\Documents\Data structure\output> & .\dellinkedlist.exe
1.Create linked list
2.delete at front
3.delete at end
4.delete an element
5.display
6.exit
enter the choice
1
Enter the element (write -999 to exit)
1
2
3
4
5
-999

1.Create linked list
2.delete at front
3.delete at end
4.delete an element
5.display
6.exit
enter the choice
2
The deleted element is 1
1.Create linked list
2.delete at front
3.delete at end
CHAT

Ask about your code

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

circularqueue.c dellinkedlist

24 ▲ 0 Data structure Debug clang: idle Live Share Debug Compile Compile & Run

Ln 133, Col 22 Spaces: 4 UTF-8 CRLF { } Go Live Prettier

PS C:\Users\88nin\OneDrive\Documents\Data structure\output> & .\dellinkedlist.exe
enter the choice
2
The deleted element is 1
1.Create linked list
2.delete at front
3.delete at end
4.delete an element
5.display
6.exit
enter the choice
3
The deleted is 5
1.Create linked list
2.delete at front
3.delete at end
4.delete an element
5.display
6.exit
enter the choice
4
Enter the value to be deleted: 3
The deleted is 3
1.Create linked list
2.delete at front
3.delete at end
4.delete an element
5.display
6.exit
enter the choice
5
CHAT

Ask about your code

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

circularqueue.c dellinkedlist

24 ▲ 0 Data structure Debug clang: idle Live Share Debug Compile Compile & Run

Ln 133, Col 22 Spaces: 4 UTF-8 CRLF { } Go Live Prettier

```
PS C:\Users\88nin\OneDrive\Documents\Data structure\output> & .\dellinkedlist.exe
4.delete an element
5.display
6.exit
enter the choice
4
Enter the value to be deleted: 3
The deleted is 3
1.create linked list
2.delete at front
3.delete at end
4.delete an element
5.display
6.exit
enter the choice
5
the elements are:-
2
4

1.create linked list
2.delete at front
3.delete at end
4.delete an element
5.display
6.exit
enter the choice
6
Exiting program
PS C:\Users\88nin\OneDrive\Documents\Data structure\output>
```

Ask about your code

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

circularqueue.c dellinkedlist.c

24 △ 0 Data structure Debug clangd: idle Live Share Debug Compile Compile & Run

Ln 133, Col 22 Spaces: 4 UTF-8 CRLF { } Go Live Prettier