Google classroom LinkedIn Facebook YouTube Music

equaledMatureBot

Run

main.c

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


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

struct node *head=NULL;

int length=0;

void insertend(int ele)
{
    struct node *newnode,*temp;
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=ele;
    newnode->next=NULL;
    if(head==NULL)
    {
        head=newnode;
        length=1;
    }
    else
    {
        temp=(struct node*)malloc(sizeof(struct node));
```

ny1/UnequaledMatureBot#main.c

Google classroom LinkedIn Facebook YouTube Music Github

equaledMatureBot Run

main.c

```
29          temp=head;
30          while(temp->next!=NULL)
31          {
32              temp=temp->next;
33          }
34          temp->next=newnode;
35          length++;
36      }
37
38  }
39
40  void insertfront(int ele)
41  {
42      struct node *temp;
43      temp=(struct node*)malloc(sizeof(struct node));
44      temp->data=ele;
45      temp->next=head;
46      head=temp;
47      length++;
48  }
49
50  void insertrandom(int ele,int pos)
51  {
52      if(pos==1)
53          insertfront(ele);
54      else if(pos>=length+1)
55          insertend(ele);
56      else
```

y1/UnequaledMatureBot#main.c

Google classroom LinkedIn Facebook YouTube Music Github

equaledMatureBot Run

main.c

```
57      {
58          struct node *inst;
59          inst=(struct node*)malloc(sizeof(struct node));
60          struct node *temp;
61          temp=(struct node*)malloc(sizeof(struct node));
62          temp=head;
63          for(int i=1;i<pos-1;i++)
64          {
65              temp=temp->next;
66          }
67          inst->data=ele;
68          inst->next=temp->next;
69          temp->next=inst;
70          length++;
71
72      }
73
74  }
75
76  void deletefront()
77  {
78      if(length==0)
79      {
80          printf("\nList is empty.\n");
81      }
82      else
83      {   struct node *temp;
84          temp=(struct node*)malloc(sizeof(struct node));
```

Meet - nyy-fmrp-bdj | Lab-7 Singly Linked List Imple

hy1/UnequaledMatureBot#main.c

Google classroom | LinkedIn | Facebook | YouTube Music | Github

equaledMatureBot | Run

main.c

```
85              temp=head;
86              head=head->next;
87              temp->next=NULL;
88              length--;
89              printf("\nThe element deleted is : %d",temp->data)
                ;
90          }
91      }

92
93      void deleteend()
94      {
95          if(length==0)
96          {
97              printf("\nList is empty.\n");
98          }
99          else
100         {   struct node *temp;
101             temp=(struct node*)malloc(sizeof(struct node));
102             temp=head;
103             while(temp->next->next!=NULL)
104             {
105                 temp=temp->next;
106             }
107             struct node *del;
108             del=(struct node*)malloc(sizeof(struct node));
109             del=temp->next;
110             temp->next=NULL;
111             length--;
```

y1/UnequaledMatureBot#main.c

Google classroom | LinkedIn | Facebook | YouTube Music | Github

qualedMatureBot

Run ▶

ain.c

```
12              printf("\nThe element deleted is : %d",del->data);
13          }
14      }
15
16      void deleterandom(int pos)
17      {
18          if(length==0)
19              printf("\nList is empty.\n");
20          else if(pos==1)
21              deletefront();
22          else if(pos>=length+1)
23              deleteend();
24          else
25          {
26              struct node *del;
27              del=(struct node*)malloc(sizeof(struct node));
28              struct node *temp;
29              temp=(struct node*)malloc(sizeof(struct node));
30              temp=head;
31              for(int i=1;i<pos-1;i++)
32              {
33                  temp=temp->next;
34              }
35              del=temp->next;
36              temp->next=del->next;
37              del->next=NULL;
38              length--;
39              printf("\nThe element deleted is : %d",del->data);
```

Lab-7 Singly Linked List Impler

adhy1/UnequaledMatureBot#main.c

Google classroom | LinkedIn | Facebook | YouTube Music | Githu

UnequaledMatureBot | Run

main.c

```
140
141         }
142
143     }
144     void display()
145     {
146         struct node *temp;
147         temp=(struct node*)malloc(sizeof(struct node));
148         temp=head;
149         if(temp==NULL)
150         {
151             printf("\n List is empty \n");
152         }
153         else
154         {
155             printf("\nThe contents of the list are :\n");
156             while(temp!=NULL)
157             {
158                 printf("%d\n",temp->data);
159                 temp=temp->next;
160             }
161         }
162
163     }
164
165     int main()
166     {
167         int choice,ele,pos;
```

Meet - nyy-fmrp-bdj | Lab-7 Singly Linked List Imple

y1/UnequaledMatureBot#main.c

Google classroom | LinkedIn | Facebook | YouTube Music | Github

qualedMatureBot | Run

ain.c

```
{
    int choice,ele,pos;
    char ch;
    do
    {
    printf("\n1. Inset at end \n2.Insert at front
    \n3.Insert at random position \n4. Display  \n5.
    Delete at front \n6.Delete at end \n7.Delete at
    random  \n8.Exit");
    printf("\nEnter your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1: printf("Enter the element to be
        inserted\n");
                scanf("%d",&ele);
                insertend(ele);
                break;
        case 2: printf("Enter the element to be
        inserted\n");
                scanf("%d",&ele);
                insertfront(ele);
                break;
        case 3: printf("Enter the element to be
        inserted\n");
                scanf("%d",&ele);
                printf("Enter the position \n");
                scanf("%d",&pos);
```

Meet - nyy-fmrp-bdj
Lab-7 Singly Linked List Imple
adhy1/UnequaledMatureBot#main.c
Google classroom
LinkedIn
Facebook
YouTube Music
UnequaledMatureBot
Run

main.c

```
                inserted\n");
185                     scanf("%d",&ele);
186                     printf("Enter the position \n");
187                     scanf("%d",&pos);
188                     insertrandom(ele,pos);
189                     break;
190             case 4: display();
191                     break;
192             case 5: deletefront();
193                     break;
194             case 6: deleteend();
195                     break;
196             case 7: printf("\nEnter the position : ");
197                     scanf("%d",&pos);
198                     deleterandom(pos);
199                     break;
200         }
201         }while(choice!=8);
202         return 0;
203 }
```