

The Time-Traveller's Messenger

locked

Problem

Submissions

Leaderboard

Discussions

Mr. X, a time traveller from the past, has just landed in the modern world. Fascinated by how people communicate, he quickly realizes that with everyone's jam-packed schedules, texting is the new calling! Determined to stay connected with his contacts, Mr. X seeks your help to create a basic chat messenger.

Features your messenger must include:

Add a new chat: So Mr. X can start new conversations.

Pin a chat: Prioritize important contacts.

Display all chats.

Unpin a chat: When a chat is no longer at the forefront.

Input Format

As per your discretion.

Constraints

No constraints

Output Format

As per your discretion.

f t in

Submissions: 66

Max Score: 0

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

[More](#)

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<ctype.h>
4 #include<strings.h>
5 #include<math.h>
6
7 typedef struct node{
8     char name[50];
9     char text[50];
10    struct node *prev;
11    struct node *next;
12 }chat;
13
```

```

14 void disp(chat *headp, chat *head);
15 chat *addchat(chat *headp, chat *head, char *n, char*t);
16 void pinchat(chat **headp, chat **head, chat *p);
17 void unpin(chat **headp, chat **head, chat *p);
18
19 int main()
20 {
21     chat* head = NULL; //pointer to first node before unpinned chats
22     chat* headp = NULL; //pointer to first node amongst pinned chats
23     int out = 0;
24     printf("*****\n");
25     printf("YOUR CHAT MESSENGER\n");
26     printf("*****\n");
27
28     while(out==0)
29     {
30         printf("Pick your action:\n");
31         printf("To DISPLAY all chats enter 0\n");
32         printf("To ADD a chat enter 1\n");
33         printf("To PIN a chat print 2\n");
34         printf("To UNPIN a chat enter 3\n");
35         int op = 0; //takes an option
36         printf("Enter option:");
37         scanf("%d",&op);
38         if(op==1)
39         {
40             char n[50] = ""; //stores name of chat
41             char t[50] = ""; //stores text in chat
42             printf("Enter name:");
43             getchar();
44             scanf("%[^\n]s",&n);
45             getchar();
46             printf("Enter text:");
47             scanf("%[^\n]s",&t);
48             head = addchat(headp,head,n,t);
49         }
50         else if(op==2)
51         {
52             printf("Enter name of contact from unpinned chats:\n");
53             getchar();
54             char cont[50] = ""; //name of contact
55             disp(headp,head);
56             scanf("%[^\n]s",&cont);
57             chat *p = head;
58             while(strcmp(p->name,cont)!=0) //moves the pointer to the specific chat
59             {
60                 p = p->next;
61             }
62
63             pinchat(&headp,&head,p);
64         }
65         else if(op==3)
66         {
67             printf("From the pinned chats enter a chat that you want to unpin\n");
68             disp(headp,head);
69             char in[50] = "";
70             getchar();
71             scanf("%[^\n]s",&in);
72             chat* point2 = headp;
73             while(strcmp(point2->name,in)!=0)
74             {
75                 point2 = point2->next;
76             }
77             unpin(&headp,&head,point2);
78         }
79         else if(op==0)
80         {
81             disp(headp,head);
82         }
83         else
84         {
85             printf("Wrong Input\n");
86         }
87     }
88 }

```

```
87     }
88 }
89
90 return 0;
91 }
92
93 void disp(chat *headp, chat *head){
94     chat *hp=headp;
95     if(head==NULL){
96         printf("Empty\n");
97     }
98     else{
99         if(headp!=NULL){
100             printf("Pinned Chats\n");
101             while (headp!=head->next && headp!=NULL){
102                 printf("Name: %s\n",headp->name);
103                 printf("Text: %s\n",headp->text);
104                 headp=headp->next;
105             }
106         }
107     }
108     printf("Unpinned Chats\n");
109     if(hp!=NULL){
110         head=head->next;
111     }
112     while(head!=NULL){
113         printf("Name: %s\n",head->name);
114         printf("Text: %s\n",head->text);
115         head=head->next;
116     }
117 }
118
119 chat *addchat(chat *headp, chat *head, char *n, char*t){
120     chat *p=(chat*)malloc(sizeof(chat));
121     p->prev=NULL;
122     p->next=NULL;
123     strcpy(p->name, n);
124     strcpy(p->text, t);
125
126     if(head==NULL){
127         head=p;
128     }
129     else if (headp==NULL && head!=NULL){
130         p->next=head;
131         head->prev=p;
132         head=p;
133     }
134     else if(headp!=NULL){
135         if(head->next==NULL){
136             p->prev=head;
137             p->next=NULL;
138             head->next=p;
139         }
140         else{
141             p->prev=head;
142             p->next=head->next;
143             head->next=p;
144             p->next->prev=p;
145         }
146     }
147     return head;
148 }
149
150 void pinchat(chat **headp, chat **head, chat *p){
151     //first detach nodes to pin
152     if(p->next==NULL){
153         if(p->prev==NULL){
154             *head=NULL;
155         }
156         else{
157             p->prev->next=NULL;
158         }
159     }
```

```
160 else if (p->prev==NULL){
161     *head=p->next;
162     (*head)->prev=NULL;
163 }
164 else{
165     p->prev->next=p->next;
166     p->next->prev=p->prev;
167 }
168 p->prev=NULL;
169 p->next=NULL;
170
171 //attach the nodes
172 if(*headp==NULL){
173     if(head==NULL){
174         *head=p;
175         *headp=p;
176     }
177     else{
178         p->next=(*head);
179         p->prev=NULL;
180         (*head)->prev=p;
181         (*head)=p;
182         (*head)->prev=NULL;
183         (*headp)=p;
184     }
185 }
186 else{
187     if((*head)->next==NULL){
188         (*head)->next=p;
189         p->next=NULL;
190         p->prev=(*head);
191         (*head)=p;
192     }
193     else{
194         p->next=(*head)->next;
195         p->prev=(*head);
196         (*head)->next=p;
197         p->next->prev=p;
198         (*head)=p;
199     }
200 }
201 }
202
203 void unpin(chat **headp, chat **head, chat *p){
204     //Pinned chat p is unpinned and placed after head
205     //if first pinned chat is chosen
206     if(p==(*headp)){
207         if((*headp)==(*head)){
208             (*headp)=NULL;
209         }
210         else{
211             //move headp to next pinned chat
212             (*headp)=p->next;
213             p->next=NULL;
214             (*headp)->next=NULL;
215             //if last pinned chat
216             if((*head)->next==NULL){
217                 (*head)->next=p;
218                 p->prev=(*head);
219             }
220             else{
221                 p->next=(*head)->next;
222                 p->prev=(*head);
223                 (*head)->next=p;
224                 p->next->prev=p;
225             }
226         }
227     }
228     //if p is last pinned chat
229     else if (p==(*head)){
230         (*head)=(*head)->prev;
231     }
232     else{
```

```
233 //remove p from linked list
234 p->prev->next=p->next;
235 p->next->prev=p->prev;
236 //if all chats are pinned
237 if((*head)->next==NULL){
238     (*head)->next=p;
239     p->prev=(*head);
240 }
241 else{
242     p->next=(*head)->next;
243     p->prev=(*head);
244     (*head)->next=p;
245     p->next->prev=p;
246 }
247 }
248 }
```

Line: 1 Col: 1

[Upload Code as File](#)

Test against custom input

Run Code

Submit Code