

```

1  #include <stdio.h>
2
3  #define MAXSIZE 20
4
5  typedef enum {
6      false = 1,
7      true = 0
8  } boolean;
9
10 typedef int ElementType;    //数组的最大容量
11 typedef struct {
12     //    ElementType *data;
13     ElementType data[MAXSIZE];
14     int length;    //当前元素的数量
15 }SqList;
16
17 void InitList(SqList *L) {
18     L->length = 0;
19     //    L->data = (ElementType *)malloc(MAXSIZE * sizeof(ElementType));
20     //    L->data[MAXSIZE];
21 }
22
23 /**
24  * 顺序输入
25  * @param list
26  * @param e
27  * @return
28  */
29 boolean ArraysInputList(SqList *list) {
30     int e;
31     if (list->length == MAXSIZE) {
32         puts("list已满!");
33     }
34     for (int i = 0; i < 4; i++) {
35         printf("输入元素");
36         scanf("%d", &e);
37         list->data[i] = e;
38         list->length++;
39     }
40     return true;
41 }
42
43 /**
44  * 按位置输入
45  * @param list
46  * @param e
47  * @param i
48  * @return
49  */
50 boolean addElement(SqList *list, int e, int i) {
51     int k;
52     if (list->length == MAXSIZE) {
53         puts("list已满");
54         return false;
55     }

```

```

56     if (i < 1 || i > list->length + 1) {
57         puts("i的值超出了list顺序表的最大长度!");
58         return false;
59     }
60     if (i <= list->length) {
61         for (k = list->length - 1; k >= i - 1; k--) {
62             list->data[k + 1] = list->data[k];
63         }
64     }
65
66     list->data[i - 1] = e;
67     list->length++;
68     return true;
69 }
70 }
71
72 /**
73  * 遍历
74  * @param list
75  * @return
76  */
77 boolean ViewList(Sqlist list) {
78
79     //从后往前
80     while (list.length != 0) {
81         printf("%d", list.data[list.length - 1]);
82         list.length--;
83     }
84     puts("\n");
85
86
87     // front --> rear
88     for (int i = 0; i < list.length; i++)
89     {
90         printf("%d", list.data[i]);
91     }
92
93     puts("\n");
94     return true;
95 }
96
97 /**
98  * 全部删除
99  * @param sqlist
100  * @return
101  */
102 boolean DeleteListEle(Sqlist *sqlist) {
103     sqlist->length = 0;
104     return true;
105 }
106
107 /**
108  * 按元素位置删除
109  * @param sqlist
110  * @param i
111  * @return
112  */
113 boolean DeletePostListEle(Sqlist *sqlist, int post) {

```

```

114     if (sqlist->length == 0) {
115         puts("list为空");
116         return false;
117     }
118     if (post > sqlist->length) {
119         puts("post大于length");
120         return false;
121     }
122
123     // for (int j = post - 1; j < sqlist->length - post - 1; j++) {
124     //     sqlist->data[j - 1] = sqlist->data[j];
125     // }
126
127     for (int j = post - 1; j < sqlist->length - post + 1; j++)
128     {
129         sqlist->data[j] = sqlist->data[j + 1];
130     }
131
132     sqlist->length--;
133     return true;
134 }
135
136 /**
137  * 按位置查找
138  * @param list
139  * @param post
140  * @return
141  */
142 boolean FindListPostEle(Sqlist list, int post)
143 {
144     if (list.length == 0)
145     {
146         puts("list为空!");
147         return false;
148     }
149     if (post > list.length) {
150         puts("post大于length");
151         return false;
152     }
153     for (int i = 0; i < list.length; i++)
154     {
155         if (i == post)
156         {
157             printf("post位置的元素为: %d\n", list.data[i - 1]);
158         }
159     }
160
161     return true;
162 }
163
164 /**
165  * 以元素查找
166  * @param sqlist
167  * @param e
168  * @return
169  */
170 boolean FindListEle(Sqlist sqlist, int e)
171 {

```

```

172
173     if (sqlist.length == 0)
174     {
175         puts("list为空! \n");
176         return false;
177     }
178     for (int i = 0; i < sqlist.length; i ++)
179     {
180         if (e == sqlist.data[i])
181         {
182             printf("元素位置为: %d\n", i + 1);
183             return true;
184         }
185     }
186     return true;
187 }
188
189 int main() {
190
191     Sqlist list;
192
193     InitList(&list);
194     puts("list顺序表初始化成功! ");
195
196     // 插入数据
197     ArraysInputList(&list);
198
199     int post;
200     printf("插入数据的位置\n");
201     scanf("%d", &post);
202     //e为要插入的元素
203     addElemtnt(&list, 6, post);
204
205     puts("遍历元素输出: \n");
206     viewList(list);
207
208     int post2;
209     puts("请输入要查找元素的位置\n");
210     scanf("%d", &post2);
211     FindListPostEle(list, post2);
212
213     int e;
214     puts("请输入要查找的元素: ");
215     scanf("%d", &e);
216     FindListELE(list, e);
217
218     puts("遍历元素输出: \n");
219     viewList(list);
220
221     int post1;
222     puts("输入删除的位置\n");
223     scanf("%d", &post1);
224     DeletePostListEle(&list, post);
225
226     puts("遍历元素输出: \n");
227     viewList(list);
228
229     puts("删除所有元素\n");

```

```
230     DeleteListEle(&list);
231
232     puts("遍历元素输出: \n");
233     viewList(list);
234
235     return 0;
236 }
```