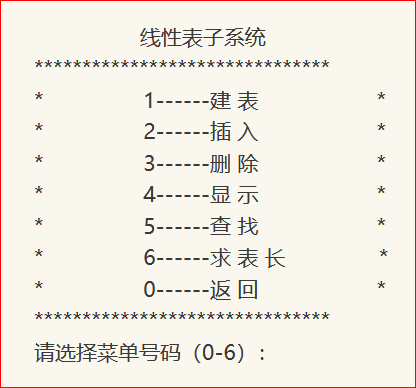
**线性表子系统**

1．实验目的   
（1）掌握线性表的特点。   
（2）掌握线性表顺序存储结构和链式存储结构的基本运算。   
（3）掌握线性表的创建、插入、删除和显示线性表中元素等基本操作。

2．实验内容   
（1）用结构体描述一个字符形的单向链表。

（2）创建线性表；在线性表中插入元素、删除元素；显示线性表中所有元素等基本操作。

（3）用if语句设计一个选择式菜单。



**附C源程序代码：**

1 #include<stdio.h>

2 #include<malloc.h>

3

4 typedef struct linknode // 结点类型定义

5 {

6 char data; // 数据域，数据类型为字符型

7 struct linknode \*next; // next为指针域

8 }linknode;

9

10 linknode \*head; // 定义 head 为结构类型指针变量

11 int n; //n为线性表的长度

12

13

14 //尾插法建表

15 // 带头结点的尾插法建立单链表

16 void createlist()

17 {

18 char x;

19 int z=1;

20 linknode \*p,\*s;

21 n=0; //线性表的长度刚开始为0

22

23 head=(linknode \*)malloc(sizeof(linknode)); // 生成头结点

24 p=head; //尾指针初值指向头结点

25

26 printf("\n\t\t请逐个输入结点。以x位结束标记!\n");

27 while(z)

28 {

29 printf("\n\t\t请输入一个字符数据，并按回车：\n");

30 scanf("\n%c",&x);

31 if(x!='x') // ‘x’为输入结束符

32 {

33 s=(linknode \*)malloc(sizeof(linknode)); // 生成新结点

34 n++; //表长加1

35 s->data=x;

36 p->next=s; // 新结点插入表尾

37 s->next=NULL;

38 p=s; // 尾指针 p 指向新的表尾

39 }

40 else

41 {

42 z=0; //遇x结束链表循环

43 }

44

45 }

46 }

47

48

49 //插入结点算法

50 void insertList(int i, char x) //i 的合法位置为：1<=i<=n

51 {

52 linknode \*s,\*p;

53 int j=0; //j用来计数

54 p=head;

55

56 printf("\t\t请输入要插的位置和数值<i x>，并按回车：\n");

57 scanf("\n%d %c",&i,&x);

58 if(i>0)

59 {

60 while(p!=NULL && j<i-1) //在结点范围内并且当j<i-1时循环执行

61 {

62 j++;

63 p=p->next; //后移指针

64 } //当j==i时循环结束

65

66 if(p!=NULL) //p还在结点范围内，肯定j==i,说明找到了

67 {

68 s=(linknode\*)malloc(sizeof(linknode)); //生成新结点

69 s->data=x;

70 s->next=p->next;

71 p->next=s; //将新结点插在位置为i的结点p的后面

72 n++; //表的长度加1

73 printf("\n\t\t插入成功");

74 }else

75 {

76 printf("\n\t\t线性表为空或插入位置超界");

77 }

78 }else

79 {

80 printf("\n\t\t抱歉!插入位置错误，请重新插入!");

81 }

82 }

83

84

85

86

87

88

89 //删除数据域为x的结点元素

90 void deleteList(char x)

91 {

92 linknode \*p,\*q;

93 q=head;

94 p=head->next; //q指向第一个元素

95 printf("\n\t\t请输入要删除的数据元素:");

96 scanf("\t\t%c",&x);

97

98 while(p!=NULL && p->data!=x)

99 {

100 q=p;

101 p=p->next;

102 } //顺链查找，直到p->data==x循环终止

103 if (p!=NULL)

104 {

105 q->next=p->next; // 如果p还在范围内，肯定p->data==x

106 free(p);

107 n--; //表的长度减1

108 printf("\n\t\t删除成功,结点%c已经被删除!",x);

109 }

110 else

111 printf("\n\t\t抱歉!没有找到您要删除的结点\n");

112

113 }

114

115 //显示线性表

116 void showlist()

117 {

118 linknode \*p=head;

119 printf("\n\t\t显示线性表的所有元素:\n\t\t");

120

121 while(p->next!=NULL) //链表不为空

122 {

123 printf("%5c",p->next->data);

124 p=p->next; //后移指针

125 }

126

127 if(head->next==NULL||p==NULL)

128 printf("\n\t\t链表为空");

129 }

130

131

132 //按序号查找

133 linknode \*queryList1(linknode \*head, int i)

134 // head接收已存在的链表的头指针

135 // i 接收要查找的结点的位置

136 //从头结点开始顺链扫描

137 {

138 linknode \*p; //用指针 p 指向当前扫描到的结点。

139 int j=1; //用 j 作统计已扫描结点数的计数器，j 的初值为 1 。

140 p=head->next; //p 的初值指链表中的第一个元素.

141

142 printf("\n\t\t请输入要查找的结点的位置，并按回车：\n");

143 scanf("\n%d",&i);

144

145 while( p->next!=NULL && j<i )

146 {

147 p=p->next;

148 j++; //当 p 扫描下一个结点时，j 自动加 1

149 }

150 if(j==i) //当 j=i时，指针 p 所指的结点就是第 i 个结点

151 {

152 printf("\n\t\t查找的此结点所在位置的值为：%c",p->data);

153 }

154

155 else

156 {

157 printf("\n\t\t未找到此节点，请确认输入是否正确!!");

158 }

159

160

161 }

162

163

164 void lengthList()

165 {

166 linknode \*p;

167 n=0;

168 p=head;

169 while(p->next!=NULL && p!=NULL)

170 {

171 n++; //表的长度加1

172 p=p->next; //后移指针

173

174 }

175 if(p->next==NULL) //如果到表尾

176 {

177 printf("\n\t\t线性表的长度为:%d",n);

178 }else

179 {

180 printf("\n\t\t链表不存在");

181 }

182

183

184 }

185

186

187 void main()

188 { int choose,i,j=1;

189 char x;

190 head = NULL;

191 while(j)

192 {

193 printf("\n\t\t\t\t 线性表子系统");

194 printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

195 printf("\n\t\t\* \t 1------建 表 \t\t\t\* ");

196 printf("\n\t\t\* \t 2------插 入 \t\t\t\* ");

197 printf("\n\t\t\* \t 3------删 除 \t\t\t\* ");

198 printf("\n\t\t\* \t 4------显 示 \t\t\t\* ");

199 printf("\n\t\t\* \t 5------查 找 \t\t\t\* ");

200 printf("\n\t\t\* \t 6------求 表 长 \t\t\* ");

201 printf("\n\t\t\* \t 0------返 回 \t\t\t\* ");

202 printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

203 printf("\n\t\t请选择菜单号码（0-6）：");

204 scanf("%d",&choose);

205 printf("\n");

206 if(choose==1)

207 {

208 createlist();

209 printf("\n\t\t\t\t 建表成功!!!\n");

210 }else if(choose==5)

211 {

212 queryList1(head, i);

213 //queryList2(head, x);

214 }else if(choose==2)

215 {

216 insertList(i, x);

217 }else if(choose==3)

218 {

219 //deleteList(i);

220 deleteList(x);

221 }else if(choose==4)

222 {

223 if(head==NULL)

224 {

225 printf("\n\t\t抱歉!线性表为空，请先建表!");

226 }else

227 {

228 showlist();

229 }

230 }else if(choose==6)

231 {

232 lengthList();

233 }else if(choose==0)

234 {

235 j=0;

236 }else

237 {

238 printf("输入错误，请重新输入!");

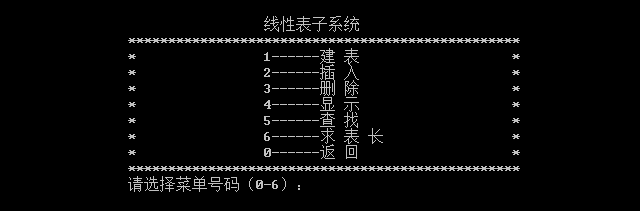
239 }

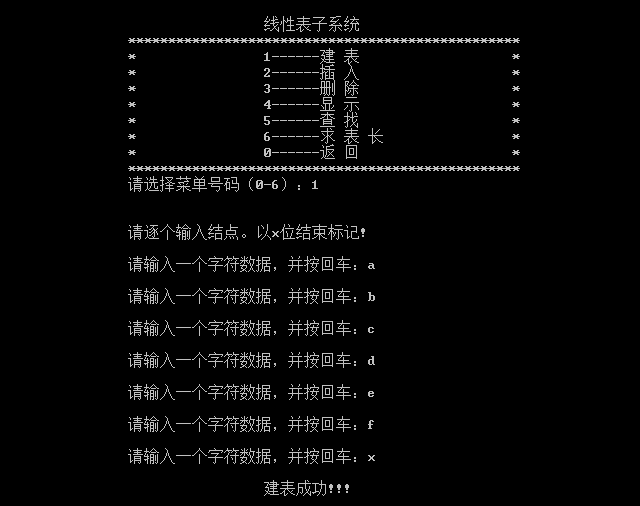
240 }

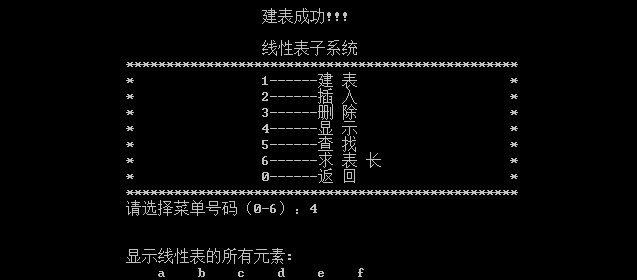
241 }

[复制代码](javascript:void(0);)

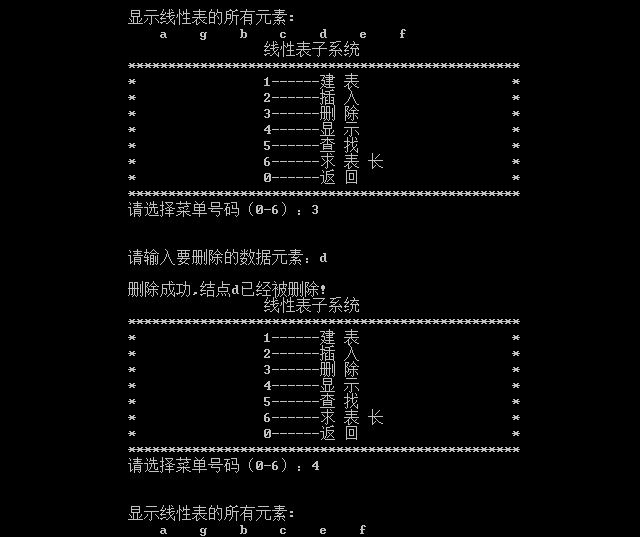
**运行结果如下：**

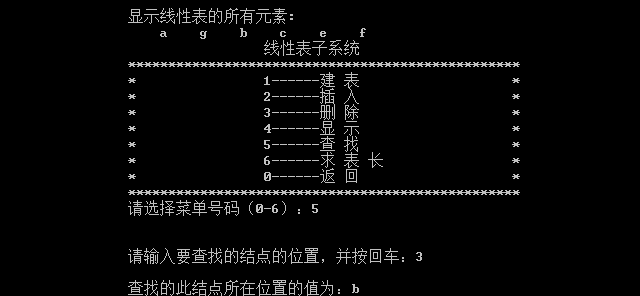


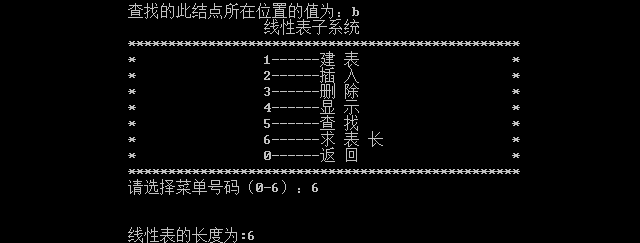


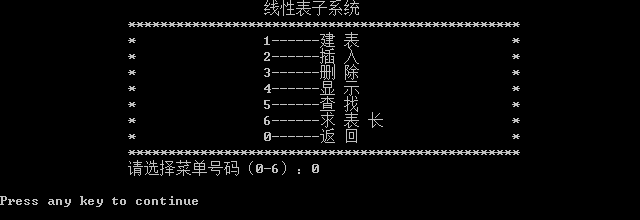












分类: [C语言](https://www.cnblogs.com/daipianpian/category/689947.html), [数据结构](https://www.cnblogs.com/daipianpian/category/688665.html)