

习题

185-5

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include<stdio.h>
3
4  typedef struct
5  {
6      int id;
7      char name[32];
8      int scores[4];
9  } Stu;
10
11  enum subjects { Chinese, Math, English, All };
12
13  static int input(Stu* students, Stu*stupt[], int n, int cnt)
14  {
15      int i;
16      for (i = cnt; i < n; i++)
17      {
18          scanf("%d", &students[i].id);
19          if (students[i].id > 0)
20          {
21              scanf("%s%d%d%d", students[i].name, &students[i].scores[0],
22                  &students[i].scores[1], &students[i].scores[2]);
23              students[i].scores[3] = students[i].scores[0] +
24                  students[i].scores[1] + students[i].scores[2];
25              stupt[i] = students + i;
26              fflush(stdin);
27          }
28          else
29          {
30              cnt = i;
31              break;
32          }
33      }
34      if (i == n)
35      {
36          cnt = n;
37          printf("overload\n");
38      }
39      return cnt;
40  }
41
42  static void addScores(Stu* students, int n, int * total_scores)
43  {
44      int i, j;
45      for (i = 0; i < n; i++)
46      {
47          for (j = 0; j < 4; j++)
48          {
49              total_scores[j] += students[i].scores[j];
50          }
51      }
52  }
```

```

49 }
50 static void findByName(char* name, Stu* students, int n)
51 {
52     int i, flag = 0;
53     Stu tmp;
54     for (i = 0; i < n; i++)
55     {
56         if (strcmp(students[i].name, name) == 0)
57         {
58             tmp = students[i];
59             if (flag == 0)
60             {
61                 printf("%8s%8s%8s%8s%8s%8s\n", "ID", "Name", "Chinese",
62 "Math", "English", "Total");
63                 flag = 1;
64             }
65             printf("%8d%8s%8d%8d%8d%8d\n", tmp.id, tmp.name,
66 tmp.scores[0], tmp.scores[1], tmp.scores[2], tmp.scores[3]);
67         }
68     }
69     if (!flag)
70     {
71         printf("No such name\n");
72     }
73 }
74 static void findByID(int id, Stu* students, int n)
75 {
76     int i, flag = 0;
77     Stu tmp;
78     for (i = 0; i < n; i++)
79     {
80         if (students[i].id == id)
81         {
82             tmp = students[i];
83             if (flag == 0)
84             {
85                 printf("%8s%8s%8s%8s%8s%8s\n", "ID", "Name", "Chinese",
86 "Math", "English", "Total");
87                 flag = 1;
88             }
89             printf("%8d%8s%8d%8d%8d%8d\n", tmp.id, tmp.name,
90 tmp.scores[0], tmp.scores[1], tmp.scores[2], tmp.scores[3]);
91         }
92     }
93     if (!flag)
94     {
95         printf("No such id\n");
96     }
97 }
98 static void showAllInfos(Stu* stupt[], int n)
99 {
100     int i;
101     Stu* tmp;
102     printf("%8s%8s%8s%8s%8s%8s\n", "ID", "Name", "Chinese", "Math",
103 "English", "Total");
104     for (i = 0; i < n; i++)
105     {
106         tmp = stupt[i];

```

```

102     printf("%8d%8s%8d%8d%8d%8d\n", tmp->id, tmp->name, tmp->scores[0],
tmp->scores[1], tmp->scores[2], tmp->scores[3]);
103     }
104 }
105 static void sort(Stu* stupt[], int n, int subj)
106 {
107     Stu* tmp;
108     int i, j;
109     for (i = 0; i < n - 1; i++)
110     {
111         for (j = n - 1; j > i; j--)
112         {
113             if (stupt[i]->scores[subj] < stupt[j]->scores[subj])
114             {
115                 tmp = stupt[i];
116                 stupt[i] = stupt[j];
117                 stupt[j] = tmp;
118             }
119         }
120     }
121 }
122
123 static void showTotalAvg(int* total_scores, int i, int cnt, char**
subjects_str)
124 {
125     printf("%8s%8s%8s\n", "subjects", "total", "average");
126     printf("%8s%8d%8.2f\n", subjects_str[i], total_scores[i], 1. *
total_scores[i] / cnt);
127 }
128
129 //int main185_5()
130 int main()
131 {
132     Stu students[100];
133     Stu* stupt[100];
134     int i;
135     int selection;
136     int cnt = 0;
137     char *subjects_str[4] = { "Chinese", "Math", "English", "All" };
138     int total_scores[4] = { 0 };
139     char input_name[100];
140     int input_id;
141
142     while (1)
143     {
144         fflush(stdin);
145         printf("### Class * Management system ###\n");
146         printf("### 1. add student info #####\n");
147         printf("### 2. total/average scores #####\n");
148         printf("### 3. find by student name #####\n");
149         printf("### 4. find by student id #####\n");
150         printf("### 5. scan all student info #####\n");
151         printf("### 6. sort by some score #####\n");
152         printf("### 7. exit management system ###\n");
153         printf("Plz input the selection: ");
154
155         scanf("%d", &selection);
156         switch (selection)

```

```

157     {
158     case 1:
159         printf("Plz input the student [id, name, Chi, Math, Eng]\n");
160         cnt = input(students, stupt, 100, cnt);
161         /*
162         123 Li 10 20 30
163         234 He 56 67 78
164         345 Ma 90 70 50
165         x
166         */
167
168         addScores(students, cnt, total_scores);
169         break;
170
171     case 2:
172         printf("\n[total / average]\n");
173         //printf("[0]Chinese;\n[1]Math;\n[2]English;\n[3]Total;\n");
174         //printf("Plz input the subject index: ");
175         //scanf("%d", &i);
176         i = 1;
177         showTotalAvg(total_scores, i, cnt, subjects_str);
178         break;
179
180     case 3:
181         printf("\n[find by name]\n");
182         //printf("Plz input the student name: ");
183         //fflush(stdin);
184         //scanf("%s", input_name);
185         strcpy(input_name, "Ma");
186         findByName(input_name, students, cnt);
187         break;
188
189     case 4:
190         printf("\n[find by id]\n");
191         //printf("Plz input the student id: ");
192         //scanf("%d", &input_id);
193         input_id = 234;
194         findByID(input_id, students, cnt);
195         break;
196
197     case 5:
198         printf("\n[show all info]\n");
199         showAllInfos(stupt, cnt);
200         break;
201
202     case 6:
203         printf("\n[sorted by scores]\n");
204         //printf("[0]Chinese;\n[1]Math;\n[2]English;\n[3]Total;\n");
205         //printf("Plz input the subject index:");
206         //scanf("%d", &i);
207         i = 0;
208         sort(stupt, cnt, i);
209         showAllInfos(stupt, cnt);
210         break;
211
212     case 7:
213         printf("\n Thanks for using.\n");
214         exit(0);

```

```

215         break;
216
217         default:
218             break;
219     }
220 }
221 }
222

```

185-6

```

1  #include<stdio.h>
2  #include<malloc.h>
3  #include<time.h>
4
5  typedef struct
6  {
7      int number;
8      int suit;
9  }Poker;
10
11  Poker cards[52];
12  char* nums[13] = { "1","2","3","4","5","6","7","8","9","10","J","Q","K" };
13  char* suits[4] = { "spades", "heart", "club", "diamond" };
14
15  void sort186_6(double randval[], Poker*index[])
16  {
17      int i, j;
18      double temp1;
19      Poker * temp2;
20      for (i = 0; i < 51; i++)
21      {
22          for (j = 51; j > i; j--)
23          {
24              if (randval[i] > randval[j])
25              {
26                  temp1 = randval[i];
27                  randval[i] = randval[j];
28                  randval[j] = temp1;
29                  temp2 = index[i];
30                  index[i] = index[j];
31                  index[j] = temp2;
32              }
33          }
34      }
35      1;
36      return;
37  }
38
39  void shuffle186_6(Poker *index[])
40  {
41      double randval[52];
42      int i, j;
43      srand(0);
44      for (i = 0; i < 52; i++)
45      {
46          randval[i] = rand();

```

```

47     }
48     sort186_6(randval, index);
49
50     printf("\n\n");
51     printf("### shuffled ###\n");
52     for (i = 0; i < 52; i++)
53     {
54         printf("%2s %s\n", nums[index[i]->number], suits[index[i]->suit]);
55     }
56     return;
57 }
58
59 void deal186_6(Poker *index[], Poker *members[4][13])
60 {
61     int i, j;
62     for (i = 0; i < 52; i++)
63     {
64         members[i % 4][i / 4] = index[i];
65     }
66
67     printf("\n\n");
68     for (i = 0; i < 4; i++)
69     {
70         printf("### member %d ###\n", i);
71         for (j = 0; j < 13; j++)
72         {
73             printf("%2s %s\n", nums[members[i][j]->number],
74             suits[members[i][j]->suit]);
75         }
76     }
77     return;
78 }
79
80 int compare186_6(Poker x, Poker y)
81 {
82     if (x.number != y.number)
83     {
84         return x.number - y.number > 0 ? 1 : -1;
85     }
86     else
87     {
88         if (x.suit == y.suit)
89         {
90             return 0;
91         }
92         return x.suit - y.suit > 0 ? 1 : -1;
93     }
94 }
95
96 //int main186_6()
97 int main()
98 {
99     Poker *index[52];
100     Poker *members[4][13];
101     char *cmp[] = { "<", "=", ">" };
102     int i, j;
103     for (i = 0; i < 52; i++)
104     {

```

```

104     cards[i].number = i / 4;
105     cards[i].suit = i % 4;
106     index[i] = &cards[i];
107 }
108 printf("### initialized ###\n");
109 for (i = 0; i < 52; i++)
110 {
111     printf("%2s %s\n", nums[index[i]->number], suits[index[i]->suit]);
112 }
113
114 shuffle186_6(index);
115
116 deal186_6(index, members);
117
118 printf("\n\n");
119 printf("### compare ###\n");
120 i = rand() % 52;
121 j = rand() % 52;
122 printf("%2s %s %s %2s %s\n", nums[index[i]->number], suits[index[i]-
>suit], cmp[compare186_6(*(index[i]), *(index[j])) + 1], nums[index[j]-
>number], suits[index[j]->suit]);
123
124 }

```

186-7

```

1  #include<stdio.h>
2  #include<malloc.h>
3
4  typedef struct Node
5  {
6      int val;
7      struct Node *next;
8  }Node;
9
10 static Node* createLList(int *a, int n)
11 {
12     Node* head = (Node*)malloc(sizeof(Node)), *node, *pre;
13     int i;
14     head->next = NULL;
15     pre = head;
16     for (i = 0; i < n; i++)
17     {
18         node = (Node*)malloc(sizeof(Node));
19         node->val = a[i];
20         node->next = NULL;
21         pre->next = node;
22         pre = pre->next;
23     }
24     return head;
25 }
26
27 static void integrate(Node* head1, Node* head2)
28 {
29     Node *pre1 = head1, *pre2 = head2, *tmp;
30     int flag;
31     int value;

```

```

32     while (pre2->next)
33     {
34         value = pre2->next->val;
35         flag = 0;
36         while (pre1->next)
37         {
38             if (value < pre1->next->val)
39             {
40                 tmp = pre2->next->next;
41                 pre2->next->next = pre1->next;
42                 pre1->next = pre2->next;
43                 pre2->next = tmp;
44                 flag = 1;
45                 break;
46             }
47             pre1 = pre1->next;
48         }
49         if (!flag)
50         {
51             tmp = pre2->next->next;
52             pre2->next->next = pre1->next;
53             pre1->next = pre2->next;
54             pre2->next = tmp;
55         }
56     }
57 }
58 static void output(Node* head)
59 {
60     while (head->next)
61     {
62         printf("%d ", head->next->val);
63         head = head->next;
64     }
65 }
66
67 //int main186_7()
68 int main()
69 {
70     int arr1[] = { 1,2,5,6 }, arr2[] = { 3,4,7,8 };
71     Node* head1 = createLList(arr1, sizeof(arr1) / sizeof(int)), *head2 =
createLList(arr2, sizeof(arr2) / sizeof(int));
72     printf("ordered linked list1: ");
73     output(head1);
74     printf("\nordered linked list2: ");
75     output(head2);
76     integrate(head1, head2);
77     printf("\nresult: ");
78     output(head1);
79     printf("\n");
80 }

```



```
Microsoft Visual Studio 调试控制台
ordered linked list1: 1 2 5 6
ordered linked list2: 3 4 7 8
result: 1 2 3 4 5 6 7 8

F:\Projects\C\Exercise\Debug\Exercise.exe (进程 12912) 已退出，返回代码为: 0。
若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...
```

186-11

```
1  #include<stdio.h>
2  #include<malloc.h>
3
4  typedef struct Node
5  {
6      int val;
7      struct Node *next;
8  }Node;
9
10 static Node* createLList(int *a, int n)
11 {
12     Node *head = NULL, *node, *pre;
13     int i;
14     head = (Node*)malloc(sizeof(Node));
15     head->next = NULL;
16     pre = head;
17     for (i = 0; i < n; i++)
18     {
19         node = (Node*)malloc(sizeof(Node));
20         node->val = a[i];
21         node->next = NULL;
22         pre->next = node;
23         pre = pre->next;
24     }
25     return head;
26 }
27
28 static void removeMinimum(Node *head)
29 {
30     Node *pre = head, *min_node, *tmp;
31     int min_val = pre->next->val;
32     min_node = pre;
33     pre = pre->next;
34
35     while (pre->next)
36     {
```

```

37         if (pre->next->val < min_val)
38         {
39             min_node = pre;
40             min_val = pre->next->val;
41         }
42         pre = pre->next;
43     }
44     tmp = min_node->next;
45     min_node->next = tmp->next;
46     tmp->next = NULL;
47 }
48
49 static void outputLList(Node* p)
50 {
51     while (p->next)
52     {
53         printf("%d ", p->next->val);
54         p = p->next;
55     }
56     printf("\n");
57 }
58 int main()
59 {
60     int arr[] = { 3,5,8,2,6,1,9,7,4 };
61     Node *head = createLList(arr, sizeof(arr) / sizeof(int));
62     outputLList(head);
63     removeMinimum(head);
64     outputLList(head);
65 }

```

Microsoft Visual Studio 调试控制台

```

3 5 8 2 6 1 9 7 4
3 5 8 2 6 9 7 4

```

F:\Projects\C\Exercise\Debug\Exercise.exe (进程 20636) 已退出，返回代码为: 0。
 若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
 按任意键关闭此窗口...

186-13

```

1  #include<stdio.h>
2  #include<malloc.h>
3  #define N 10
4
5  typedef struct Text
6  {

```

```

7     char val;
8     struct Text *next;
9 }Text;
10
11 static Text* createLList(char *source, int n)
12 {
13     Text *head = NULL, *node, *pre = NULL;
14
15     int i;
16     for (i = 0; i < n; i++)
17     {
18         node = (Text*)malloc(sizeof(Text));
19         node->val = source[i];
20         node->next = NULL;
21         if (!head)
22         {
23             head = pre = node;
24         }
25         else
26         {
27             pre->next = node;
28             pre = pre->next;
29         }
30     }
31     pre->next = head;
32     return head;
33 }
34
35 static char* generateCipher(char *source, int key, int n)
36 {
37     Text* head = createLList(source, n);
38     Text *pre = NULL, *cur = head;
39     char *res = (char*)malloc(sizeof(char) * (n + 1));
40     int i, j;
41     while (cur->next != head)
42     {
43         cur = cur->next;
44     }
45     pre = cur;
46     cur = head;
47     for (i = 0; i < n; i++)
48     {
49         j = key;
50         while (--j)
51         {
52             pre = cur;
53             cur = cur->next;
54         }
55         res[i] = cur->val;
56         pre->next = cur->next;
57         cur->next = NULL;
58         cur = pre->next;
59     }
60     res[n] = '\0';
61     return res;
62 }
63
64 int main186_13()

```

```

65 //int main()
66 {
67     int key = 2;
68     char source[N+1] = "0123456789";
69     char *cipher = generateCipher(source, key, N);
70     printf(" input:%s [key=%d]\n", source, key);
71     printf("output:%s", cipher);
72 }

```

```

Microsoft Visual Studio 调试控制台
input:0123456789 [key=2]
output:1357926084
F:\Projects\C\Exercise\Debug\Exercise.exe (进程 22176) 已退出，返回代码为: 0。
若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动
关闭控制台”。
按任意键关闭此窗口...

```

186-14

```

1  #include<stdio.h>
2  #include<malloc.h>
3
4  typedef struct Node_186_14
5  {
6      int val;
7      struct Node_186_14 *next;
8  }Node;
9
10 static Node* createLList(int *a, int n)
11 {
12     int i;
13     Node* head = (Node*)malloc(sizeof(Node)), *pre = head, *node;
14     head->next = NULL;
15     for (i = 0; i < n; i++)
16     {
17         node = (Node*)malloc(sizeof(Node));
18         node->val = a[i];
19         node->next = NULL;
20         pre->next = node;
21         pre = pre->next;
22     }
23     return head;
24 }
25 static void outputLList(Node* set)
26 {
27     Node* cur = set;

```

```

28     while (cur->next)
29     {
30         printf("%d ", cur->next->val);
31         cur = cur->next;
32     }
33     printf("\n");
34 }
35 static void sumSet(Node *set1, Node* set2)
36 {
37     Node *cur1, *cur2 = set2, *tmp;
38     int value;
39     int flag;
40     while (cur2->next)
41     {
42         value = cur2->next->val;
43         cur1 = set1;
44         flag = 0;
45         while (cur1->next)
46         {
47             if (cur1->next->val == value)
48             {
49                 tmp = cur2->next;
50                 cur2->next = tmp->next;
51                 tmp->next = NULL;
52                 flag = 1;
53                 break;
54             }
55             else if (cur1->next->val > value)
56             {
57                 tmp = cur2->next->next;
58                 cur2->next->next = cur1->next;
59                 cur1->next = cur2->next;
60                 cur2->next = tmp;
61                 flag = 1;
62                 break;
63             }
64             cur1 = cur1->next;
65         }
66         //尾结点
67         if (!flag)
68         {
69             tmp = cur2->next->next;
70             cur2->next->next = cur1->next;
71             cur1->next = cur2->next;
72             cur2->next = tmp;
73             flag = 1;
74         }
75         //cur2 = cur2->next;
76     }
77 }
78 static void minusSet(Node *set1, Node* set2)
79 {
80     Node *cur1 = set1, *cur2 = set2, *tmp;
81     int value;
82     while (cur2->next)
83     {
84         value = cur2->next->val;
85         cur1 = set1;

```

```

86     while (cur1->next)
87     {
88         if (cur1->next->val == value)
89         {
90             tmp = cur1->next;
91             cur1->next = tmp->next;
92             tmp->next = NULL;
93         }
94         else if (cur1->next->val > value)
95         {
96             break;
97         }
98         if (!cur1->next)
99         {
100             break;
101         }
102         cur1 = cur1->next;
103     }
104     cur2 = cur2->next;
105 }
106 }
107 static void unionSet(Node *set1, Node* set2)
108 {
109     Node *cur1 = set1, *cur2 = set2, *tmp;
110     int flag;
111     int value;
112     while (cur1->next)
113     {
114         cur2 = set2;
115         value = cur1->next->val;
116         flag = 0;
117         while (cur2->next)
118         {
119             if (value == cur2->next->val)
120             {
121                 flag = 1;
122                 break;
123             }
124             cur2 = cur2->next;
125         }
126         if (!flag)
127         {
128             tmp = cur1->next;
129             cur1->next = tmp->next;
130             tmp->next = NULL;
131         }
132         else
133         {
134             cur1 = cur1->next;
135         }
136     }
137 }
138 }
139
140 //int main_186_14()
141 int main()
142 {
143     int arr1[] = { 2,3,5,6 }, arr2[] = { 3,4,6,8 };

```

```

144     Node *set1, *set2;
145
146     printf("### setSum ###\n");
147     set1 = createLList(arr1, sizeof(arr1) / sizeof(int));
148     set2 = createLList(arr2, sizeof(arr2) / sizeof(int));
149     outputLList(set1);
150     outputLList(set2);
151     sumSet(set1, set2);
152     outputLList(set1);
153
154     printf("\n### setDiff ###\n");
155     set1 = createLList(arr1, sizeof(arr1) / sizeof(int));
156     set2 = createLList(arr2, sizeof(arr2) / sizeof(int));
157     outputLList(set1);
158     outputLList(set2);
159     minusSet(set1, set2);
160     outputLList(set1);
161
162     printf("\n### setUnion ###\n");
163     set1 = createLList(arr1, sizeof(arr1) / sizeof(int));
164     set2 = createLList(arr2, sizeof(arr2) / sizeof(int));
165     outputLList(set1);
166     outputLList(set2);
167     unionSet(set1, set2);
168     outputLList(set1);
169 }

```

```

Microsoft Visual Studio 调试控制台
### setSum ###
2 3 5 6
3 4 6 8
2 3 4 5 6 8

### setDiff ###
2 3 5 6
3 4 6 8
2 5

### setUnion ###
2 3 5 6
3 4 6 8
3 6

F:\Projects\C\Exercise\Debug\Exercise.exe (进程 24624) 已退出，返回代码为: 0。
若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动
关闭控制台”。
按任意键关闭此窗口...

```