

20160311学习笔记

stuuct_array

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  void main(){
4.      struct student {
5.          int num;
6.          char name[20];
7.      }stu2[] = {{1000,"zhangsan"},{1001,"wangwu"}};
8.      int i;
9.      struct student *p;//结构体指针
10.     p = stu;//p+1 就偏移到 stu[1]
11.     for(i=0;i<2;i++){
12.         printf("the num = %d,name = %s\n",(p+i)->num,(p+i)->name);
13.     }
14.     system("pause");
15. }
```

```
1.  list.h
2.  #include <stdio.h>
3.  #include <stdlib.h>
4.  #include <string.h>
5.  typedef struct student {
6.      int num;
7.      struct student *next;
8.  }stu,*psdu;
9.  void link_head_insert(pstu*,psdu*,int);
10. void link_tail_insert(pstu*,psdu*,int);
11. void link_sort_insert(pstu*,psdu*,int);
12. void link_show(pstu);
13. void link_delate(pstu*,psdu*,int);
14. void link_modify(pstu,int,int);
15. int link_search(pstu,int);
```

```
1.  #include "list.h"
2.  void main(){
3.      pstu phead,ptail;
4.      phead =NULL;
5.      ptail = NULL;
6.      int num;
7.      while(scanf("%d",&num)!=EOF){
8.          //link_head_insert(&phead,&ptail,num);
9.          // link_tail_insert(&phead,&ptail,num);
10.         link_sort_insert(&phead,&ptail,num);
11.     }
12.     link_show(phead);
13.     scanf("%d",&num);
14.     link_show(phead);
15.     system("pause");
16. }
```

```

1.  #include "list.h"
2.  //头插法新增链表
3.  void link_head_insert(pstu *phead ,pstu *ptail ,int n){
4.      pstu pnew = (pstu)malloc(sizeof(stu));
5.      memset(pnew,0,sizeof(stu)); //对申请的结构体空间全部赋0
6.      pnew ->num = n;
7.      if(*phead == NULL){
8.          *phead = pnew;
9.          *ptail = pnew;
10.     }else{
11.         pnew->next = *phead;
12.         *phead = pnew;
13.     }
14. }
15. //尾插法新增节点
16. void link_tail_insert(pstu *phead ,pstu *ptail ,int n){
17.     pstu pnew = (pstu)malloc(sizeof(stu));
18.     memset(pnew,0,size(stu)); //对申请的结构体空间全部赋0
19.     pnew ->num = n;
20.     if(*ptail == NULL){
21.         *phead = pnew;
22.         *ptail = pnew;
23.     }else{
24.         (*ptail)->next = pnew; //将新增节点地址赋给原有尾节点的next
25.         *ptail = pnew; //
26.     }
27. }
28. //打印节点
29. void link_show(pstu phead){
30.     pstu pcur = phead;
31.     while(pcur!=NULL){
32.         printf("%3d",pcur->num);
33.         pcur = pcur->next;
34.     }
35.     printf("\n");
36. }
37. //有序插入
38. void link_sort_insert(pstu *phead ,pstu *ptail ,int n){
39.     pstu ppre;
40.     pstu pcur;
41.     pstu pnew = (pstu)malloc(sizeof(stu));
42.     memset(pnew,0,size(stu)); //对申请的结构体空间全部赋0
43.     pnew ->num = n;
44.     if(*pheadl == NULL){
45.         *head = pnew;
46.         *ptail = pnew
47.     }else if((*phead)->num>n){
48.         pnew->pnext = *head;
49.         *phead = pnew;
50.     }else if((*ptail)->num<n){
51.         (*ptail)->next = pnew;
52.         *ptail = pnew;
53.     }else{
54.         pcur =phead;
55.         while(pcur !=NULL){

```

```

56.         if(pcur->num>n){
57.             ppre->next = pnew;
58.             pnew->next = pcur;
59.             break;
60.         }
61.         ppre = pcur;
62.         pcur = pcur->next;
63.     }
64. }
65. }
66.
67. void link_delete(pstu *phead, pstu *ptail, int n)
68. {
69.     pstu ppre = NULL;
70.     pstu pcur;
71.     if(*phead == NULL) {
72.         printf("the link is empty");
73.     }else{
74.         pcur = *phead;
75.         while(pcur != NULL)
76.         {
77.             if(pcur -> num ==n)
78.             {
79.                 if(ppre ==NULL)
80.                 {
81.                     *phead = pcur -> pnext;
82.                 }else{
83.                     ppre -> pnext = pcur -> pnext;
84.                 }
85.                 free(pcur);
86.                 pcur = NULL;
87.                 return;
88.             }
89.             ppre = pcur;
90.             pcur = pcur -> pnext;
91.         }
92.         printf("THis Node doesn't exist!\n");
93.     }
94. }
95.
96. int link_search(pstu phead,int n){
97.     if(phead == NULL){
98.         return 0;
99.     }else{
100.         while(phead != null){
101.             if(phead ->num == n){
102.                 retuen 1;
103.             }
104.             phead = phead ->next;
105.         }
106.     }
107.     return 0;
108. }
109.
110. void link_modify(pstu phead,int m ,int n){
111.     int flag = 0;

```

```
112.         while(phead != null){
113.             if(phead ->num == m){
114.                 phead->num = n;
115.                 flag++;
116.             }
117.             phead = phead ->next;
118.         }
119.         return flag;}
120.
```

1 4 10 17 28 插入 15 怎么插

有序插入

union 成员公用空间

enum 每一个对应的值都是常量值