20160311学习笔记

stuuct_array

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

```c
list.h
#inlclude <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef  struct student {
    int num;
    struct student *next;
}stu,*psdu;
void link_head_insert(pstu*,pstu*,int);
void link_tail_insert(pstu*,pstu*,int);
void link_sort_insert(pstu*,pstu*,int);
void link_show(pstu);
void link_delate(pstu*,pstu*,int);
void link_modify(pstu,int,int);
int link_search(pstu,int);
```

```c
#include "list.h"
void main(){
    pstu phead,ptail;
    phead =NULL;
    ptail = NULL;
    int num;
    while(scanf("%d",&num)!=EOF){
        //link_head_insert(&phead,&ptail,num);
       // link_tail_insert(&phead,&ptail,num);
        link_sort_insert(&phead,&ptail,num);
    }
    link_show(phead);
    scanf("%d",&num);
    link_show(phead);
    system("pause");
}
```

```c
#include "list.h"
//头插法新增链表
void link_head_insert(pstu *phead ,pstu *ptail ,int n){
    pstu pnew = (pstu)malloc(sizeof(stu));
    memset(pnew,0,sizeof(stu));//对申请的结构体空间全部赋0
    pnew ->num = n;
    if(*phead == NULL){
        *phead = pnew;
        *phead = pnew;
    }else{
        pnew->next = *phead;
        *phead = pnew;
    }
}
//尾插法新增节点
void link_tail_insert(pstu *phead ,pstu *ptail ,int n){
    pstu pnew = (pstu)malloc(sizeof(stu));
    memset(pnew,0,size(stu));//对申请的结构体空间全部赋0
    pnew ->num = n;
    if(*ptail == NULL){
        *phead = pnew;
        *ptail = pnew;
    }else{
        (*ptail)->next = pnew;//将新增节点地址赋给原有尾节点的next
        *ptail = pnew;//
    }
}
//打印节点
void link_show(pstu phead){
    pstu pcur = phead;
    while(pcur!=NULL){
        printf("%3d",pcur->num);
        pcur = pcur->next;
    }
    printf("\n');
}
//有序插入
void link_sort_insert(pstu *phead ,pstu *ptail ,int n){
     pstu ppre;
    pstu pcur;
    pstu pnew = (pstu)malloc(sizeof(stu));
    memset(pnew,0,size(stu));//对申请的结构体空间全部赋0
    pnew ->num = n;
    if(*pheadl == NULL){
        *head = pnew;
        *ptail = pnew
    }else if((*phead)->num>n){
         pnew->pnext = *head;
        *phead = pnew;
    }else if((*ptail)->num<n){
         (*ptail)->next = pnew;
        *ptail = pnew;
    }else{
        pcur =phead;
        while(pcur !=NULL){
```

```
            if(pcur->num>n){
                ppre->next = pnew;
                pnew->next =pcur;
                break;
            }
            ppre = pcur;
            pcur = pcur->next;
        }
    }
}

void link_delete(pstu *phead, pstu *ptail, int n)
{
    pstu ppre = NULL;
    pstu pcur;
    if(*phead == NULL) {
        printf("the link is empty");
    }else{
        pcur = *phead;
        while(pcur != NULL)
        {
            if(pcur -> num ==n)
            {
                if(ppre ==NULL)
                {
                    *phead = pcur -> pnext;
                }else{
                    ppre -> pnext = pcur -> pnext;
                }
                free(pcur);
                pcur = NULL;
                return;
            }
                ppre = pcur;
                pcur = pcur -> pnext;
        }
        printf("THis Node doesn't exist!\n'");
    }
}

int link_search(pstu phead,int n){
    if(phead == NULL){
        return 0;
    }else{
        while(phead != null){
            if(phead ->num == n){
                retuen 1;
            }
            phead = phead ->next;
        }
    }
    return 0;
}

void link_modify(pstu phead,int m ,int n){
    int flag = 0;
```

```
      while(phead != null){
             if(phead ->num == m){
                 phead->num = n;
                 flag++;
             }
             phead = phead ->next;
         }
     return flag;}
```

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