



---

# 数据结构-实验三





# 实验三：链式存储

---

## 一、实验目的

- 1、复习结构体、指针；
- 2、掌握链表的创建、遍历等操作；
- 3、了解函数指针。



# 引例

```
typedef struct score_grade
{
    int score;
    char grade;
    struct score_grade *next; } student;

student* Input_Score(int n)
//创建链表存储n个学生的分数与等级，通过键盘输入分数
{
    int i; student *stu, *p;
    for(i=0, stu=NULL; i<n; i++)
    {
        p=(student*)malloc(sizeof(student));
        printf("请输入第%d个学生的分数(0—100): \n",i+1);
        scanf("%d",&p->score);
        while(p->score<0 || p->score>100)
        {
            printf("请重新输入第%d个学生的分数(0—100): \n",i+1);
            scanf("%d",&p->score);
        }
        p->next=stu;
        stu=p;
    }
    return stu;
}
```



# 引例

```
void Visit(student *stu, void (*fun)())  
{ student *p;  
  for(p=stu; p!=NULL; p=p->next) (*fun)(p);  
}
```

```
void Score_Grade(student *p)  
//指针p所指结点存储学生的分数与等级，转换分数为等级  
{ switch(p->score/10)  
  { case 10:  
    case 9: p->grade='A'; break;  
    case 8:  
    case 7: p->grade='B'; break;  
    case 6: p->grade='C'; break;  
    default: p->grade='D'; }  
}
```



# 引例

```
void Output_Score_Grade(student *p)
//指针p所指结点存储学生的分数与等级，输出分数与等级
{   printf("%d\t%c\n", p->score, p->grade);   }

int main()
{   int n;
    student *stu;
    printf("请输入学生人数（1-100）：\n");
    scanf("%d", &n);
    while(n<1 || n>100)
    {   printf("请重新输入学生人数（1-100）：\n");
        scanf("%d", &n);   }
    stu=Input_Score(n);
    Visit(stu,Score_Grade);
    printf("各个学生的分数与等级如下：\n分数\t等级\n");
    Visit(stu,Output_Score_Grade);
}
```



# 引例

```
typedef struct Lnode { // 结点类型
    ElemType data;      // 数据域
    struct Lnode *next; // 指针域
} Lnode, *Link;
```

```
typedef struct LinkList { // 链表类型
    Link head, tail; // 头和尾结点指针
    int len;          // 当前长度
} LinkList;
```

```
typedef struct score_grade {
    int score;
    char grade;
} student;
```

```
typedef struct stu_node { // 学生结点
    student data;          // 学生域
    struct stu_node *next; // 指针域
} stu_node, *stu_link;
```

```
typedef struct stu_list { // 学生链表
    stu_link head, tail; // 头和尾结点指针
    int len;              // 学生人数
} stu_list;
```



# 引例

---

思考1：链式存储与顺序存储的区别是什么？

思考2：链表创建、遍历等操作的特点是什么？

思考3：函数指针的作用是什么？



# 实验三：链式存储

---

## 二、实验内容

- 1、（必做题）每个学生的成绩信息包括：学号、语文、数学、英语、总分、加权平均分；**采用链表**存储若干学生的成绩信息；输入学生的学号、语文、数学、英语成绩；计算学生的总分和加权平均分（语文占30%，数学占50%，英语占20%）；输出学生的成绩信息。





# 实验三：链式存储

---

## 二、实验内容

- 2、（必做题）可以在链表末尾追加新学生的成绩信息；可以根据学号，删除该学生的成绩信息。
- 3、（选做题）可以根据学号或总分，升序排序学生的成绩信息。
- 4、（选做题）数据结构MOOC的第二周编程实训题第2题：2-2 单链表逆转（在拼题A网站 <https://pintia.cn/> 上完成）