

《数据结构》实验报告 2

学号: _____ 姓名: _____ 班级: _____ 成绩: _____

实验名称: 线性表的顺序及链式存储

实验地点: 数学系机房

所使用的工具软件及环境: VC++

一、 实验目的:

- 1、通过本实验, 掌握线性表的顺序存储和链接存储的表示方法
- 2、并能在线性结构的基本操作的基础上, 设计合适的算法解决实际问题

二、 评分标准:

1. 评分成绩为 A,B,C,D,E 五档, 满分为 A
 - 1) 单选题: 每错 3 个小题, 总分降一档。
 - 2) 填空题: 每错 2 个小题, 总分降一档。
 - 3) 程序填空: 每错 1 空, 总分降一档。

三、 实验内容(1):

(一) 单选题

1、在一个长度为 n 的顺序存储的线性表中, 向第 i 个元素($1 \leq i \leq n+1$)之前插入一个新元素时, 需要从后向前依次移_____个元素。

A $n-i$ B $n-i+1$ C $n-i-1$ D i

2、除了_____, 其它任何指针都不能在算法中作为常量出现, 也无法显示。

A 头指针 B 尾指针 C 指针型变量 D 空指针

3、在一个长度为 n 的线性表中顺序查找值为 x 的元素时, 查找成功的平均查找长度 (即 x 同元素的平均比较次数, 假定查找每个元素的概率都相等) 为_____。

A n B $n/2$ C $(n+1)/2$ D $(n-1)/2$

4、在一个单链表 HL 中, 若要向表头插入一个自由指针 p 指向的结点, 则执行_____。

A $HL=p; p->next=HL;$ B $p->next=HL; HL=p;$
C $p->next=HL; p=HL;$ D $p->next=HL->next; HL->next=p;$

5、在一个单链表 HL 中, 若要在指针 q 所指结点的后面插入一个自由指针 p 所指向的结点, 则执行_____。

A $q->next=p->next; p->next=q;$ B $p->next=q->next; q=p;$
C $q->next=p->next; p->next=q$ D $p->next=q->next; q->next=p;$

6、在一个单链表 HL 中, 若要删除由指针 q 所指向结点的后继结点, 则执行_____。

A $p=q->next; p->next=q->next;$ B $p=q->next; q->next=p;$
C $p=q->next; q->next=p->next;$ D $q->next=q->next->next; q->next=q;$

7、对于顺序表, 下面说法错误的是_____。

- A 顺序表是用一维数组实现的线性表, 数组的下标可以看成是元素的绝对地址
B 顺序表的所有存储结点按相应数据元素间的逻辑关系决定的次序依次排列
C 顺序表的特点是: 逻辑结构中相邻的结点在存储结构中仍相邻

D 顺序表的特点是:逻辑上相邻的元素, 存储在物理位置也相邻的单元中;

(二) 填空题

1. 线性表典型的基本运算包括: _____、_____、_____、_____、_____、_____等六种。
2. 对于一个长度为 n 的顺序存储的线性表, 在表头插入元素的时间复杂度为_____, 在表尾插入元素的时间复杂度为_____。
3. 线性结构的基本特征是:若至少含有一个结点, 则除起始结点没有直接_____外, 其他结点有且仅有一个直接_____; 除终端结点没有直接_____外, 其它结点有且仅有一个直接_____。
4. 在线性表的单链接存储中, 若一个元素所在结点的地址为 p , 则后继结点的地址为_____, 若假定 p 为一个数组 a 中的下标, 则其后继结点的下标为_____。
5. 在循环单链接表中, 最后一个结点的指针域指向_____结点。
6. 在双向链接表中每个结点包含有两个针域, 一个指向其_____点, 另一个指向其_____结点。
7. 在循环双向链接表中表头结点的左指针域指向_____结点, 最后一个结点的右指针域指向_____结点。
8. 在由数组 a 中元素结点构成的单链表中, 在下标为 i 的结点的后面插入一个下标为 j 的结点时, 需要进行的操作为_____。

(三) 实验内容(2): 顺序表的实现--程序填空,

- (1)创建顺序表
- (2)在顺序表的第 i 位插入元素
- (3)删除顺序表的第 i 个元素
- (4)输出顺序表
- (5)判断顺序表是否为空
- (6)判断顺序表是否满
- (7)求顺序表第 i 个元素的值
- (8)查找值为 x 的元素

```
#include<stdio.h>
#include<malloc.h>
#define MaxSize 100

typedef int dataType;
typedef struct {
    dataType data[MaxSize];
    int size;
} SqList;

//创建顺序表
SqList* CreateList() {
```

```

    SqList *t = (SqList *)malloc(sizeof(SqList));
    t->size = 0;
    return t;
}

```

//在顺序表的第 k 个位置插入元素 x

```

void Insert(SqList *l, int k, dataType x) {
    if (k < 1 || k > l->size + 1) exit(1);
    if (l->size == MaxSize) exit(1);
    for (int i = l->size; i >= k; i--)
        l->data[i] = l->data[i-1];
    l->data[k-1] = _____;
    l->size++;
}

```

//删除顺序表的第 k 个元素

```

void Delete(SqList *l, int k) {
    if (k < 1 || k > l->size) exit(1);
    for (int i = k; i < l->size; i++)
        l->data[i-1] = _____;
    l->size--;
}

```

//判断顺序表是否为空

```

int Empty(SqList *l) {
    return l->size == 0;
}

```

//判断顺序表是否满

```

int Full(SqList *l) {
    return l->size == _____;
}

```

//求顺序表第 i 个元素的值

```

dataType GetData(SqList *l, int i) {
    if (i < 1 || i > l->size) exit(1);
    return l->data[ _____];
}

```

```

//查找值为 x 的元素
int Locate(SqList *l, dataType x) {
    for (int i = 0; i < l->size; i++)
        if (l->data[i]_____x)
            return i + 1;
    return 0;
}

//输出顺序表
void Print(SqList *l) {
    for (int i = 0; i < l->size; i++)
        printf("%d ", l->data[i]);
    printf("\n");
}

int main() {
    SqList *pl = CreateList();
    Insert(pl, 1, 10);
    Insert(pl, 1, 20);
    Delete(pl, 2);
    Insert(pl, 1, 30);
    Insert(pl, 1, 40);
    Print(pl);
    printf("%d", GetData(pl, 2));
}

```

四、 评语：

任课教师：赵宏庆 成绩： _____ 年 ____ 月 ____ 日