

一. 选择题

BBBAC DACCD BDBC

二. 填空题

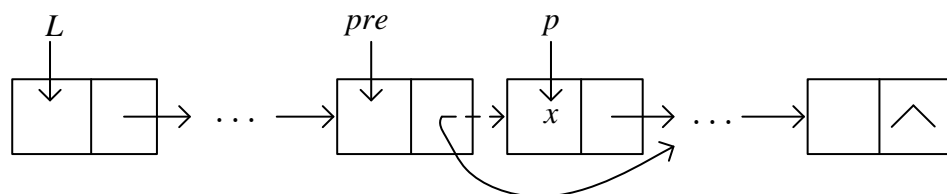
- ① p->next
② s->data
③ t
- ① (Node *)malloc(sizeof(Node))
② tail = p
③ p = p->next
④ L->next = q
⑤ q = q->next
- 8
- CHAR

三. 算法设计题

- 解析：先在有序顺序表 L 中找到适当的位置，然后后移元素空出一个位置，再将 x 插入. 对应算法如下：

```
int insert(List L, ElementType x)
{
    int i=0, j;
    if(L->Last == MAXSIZE-1)           // 表空间已满，不能插入
    {
        printf("表满");
        return 0;
    }
    while(i<=L->Last && x>=L->Data[i]) //查找插入位置 i
        i++;
    for(j=L->Last; j>=i; j--)
        L->Data[j+1]=L->Data[j];       //移出插入 x 的位置
    L->Data[i] = x;                     //x 插入位置 i
    L->Last++;                          //Last 仍指向最后元素
    return 1;
}
```

- 解析：如图所示，假定 L 为带头结点的单链表，用 pre、p 遍历整个单链表，pre 指向 p 的前驱节点，p 用于查找第一个值为 x 的节点，当找到后将 p 节点删除，返回 1；否则返回 0. 对应算法如下：



```

int delete(List L, ElementType x)    //默认 L 有头结点
{
    Position pre,p;
    pre = L;
    p = pre->Next;

    while (p != NULL && p->Data != x) {
        pre = p;
        p = p->Next;           //pre、p 同步后移一个节点
    }
    if (p != NULL) {           //找到值为 x 的 p 节点
        pre->Next = p->Next;
        free(p);
        return 1;
    } else {
        return 0;              //未找到值为 x 的 p 节点
    }
}

```