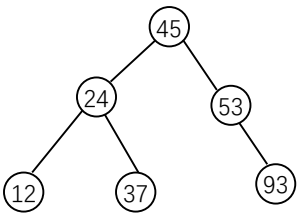
一、解答题

1 请计算二叉排序树的平均查找长度 ASL。



2 已知一棵二叉树的中序遍历序列为 cbdagf，后序遍历序列为 cdbgfa，画出该二叉树。

3 请证明深度为 k 的二叉树至多有 2^k-1 个结点（$k \geqslant 1$）。

4 证明对于任何一棵二叉树，如果其终端结点数为 n_0，度为 2 的结点数为 n_2，则 $n_0=n_2+1$。

5 证明：一棵有 n 个叶子结点的赫夫曼树共有 2n-1 个结点。

6 已知序列（10，18，4，15，12），用快速排序算法对其升序排序，请写出每一趟排序的结果。

7 已知序列（10，18，4，3，6，12），用快速排序算法对其升序排序，请写出每一趟排序的结果。

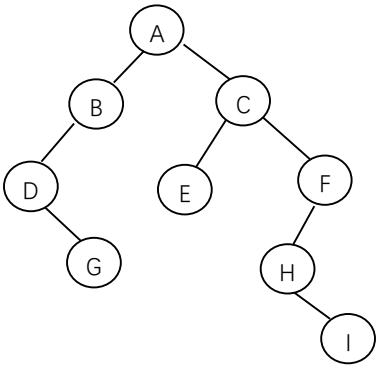
8 设查找关键字序列为{45，24，53，45，12}，请画出所对应的二叉排序树（要求过程，否则不得分）。

9 设查找关键字序列为{34，48，66，25，18，39}，请构造对应的二叉排序树（要求过程，否则不得分）。

10 写出 f1(4)的运行结果。

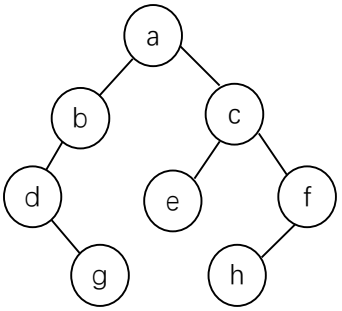
```
void f1(int n)
{   int i;
    if(n!=0)
    {   for(i=1;i<=n;i++)
            printf("%d",i);
        printf("\n");
        f1(n-1); }
}
```

11 将二叉树转换为对应的森林。



12 已知关键字序列（10，4，18，3，6，12），用简单选择排序算法对其进行升序排序，请写出每一趟排序的结果。

13 根据下面的二叉树，写出三种遍历方案下的遍历序列。



14 已知序列（10，18，4，15，12，17），用直接插入排序算法对其进行升序排序，请写出每一趟排序的结果。

二、算法填空题

1 下面算法的功能是检验算术表达式中括号是否匹配，请将算法补充完整。

设栈结构类型为 sqstack，且已提供的操作有:

出栈操作　char pop(sqstack *sp)

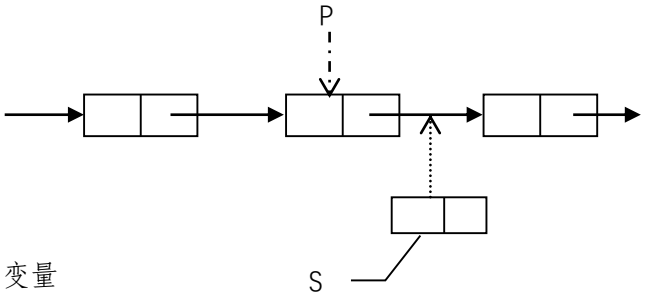
入栈操作　int push(sqstack *sp,char s)

判栈空操作　int empty(sqstack *sp)

```
int isperfact(char str[])
{   int i;
    sqstack *sp;
    Initstack(sp);
    for(i=0;str[i];i++)
    {   if(str[i]==' (' )
            ____①____;
        else if(____②____)
            if(!empty(sp))
                ____③____;
            else
                return false;
```

```
        }
        if(      ④      )
            return false;
        ____⑤____;
    }
```

2 在一个单链表中的 p 所指结点之前插入一个 s 所指的新结点，算法可分两步进行，第一步把*s 插入在*p 之后，第二步交换*s 和*p 的数据。请填空。




```
        /*结点类型定义*/
        typedef struct Lnode
        {   int data;
            ____①____ *next;
        }Llinklist;
        /*部分算法*/
        ……
        int temp;//作为交换时的中间变量
        ____②____;
        ____③____;
        temp =p->data;
        ____④____;
        ____⑤____;
```

3 该算法的功能是实现将带头结点的单链表就地置逆，请将算法补充完整。

```
typedef struct node
    {   datatype data;
        ____①____ ;
    }____②____ ;
    void reverse(node *first)
    {   node *p,*r,*pre; //p 指针指向当前处理的结点,pre 指针指向 p 的前趋结点
        p=____③____;                                    r 指针指向 p 的后继结点
        pre=NULL;
        while(  ____④____  )
        {   r=____⑤____;
            p->next=pre;
            pre=p;
            p=r;
        }
        first->next=pre;
    }
```

4 折半插入排序算法是从减少比较次数角度来改进直接插入排序算法的，请将算法补充完整。

```
    Void BInsertSort(Sqlist &L)//L 为排序表
    {   for(i=2;i<=L.length;++i)
```

```
        {   L.r[0]=L.r[i];
            Low=1;
            ____①____ ;
            While( ____②____ )
            {  ____③____;
               if(LT(L.r[0].key,L.r[m].key)
                  high=m-1;
               else
                  ____④____;
            }
            for(j=i-1; j>=high+1;--j)
               L.r[j+1]=L.r[j];
            L.r[high+1]=____⑤____;
        }
    }
```

5 以下是在中序线索二叉树中查找结点*p 的中序前趋算法，请填空。

```
        bithptr  *INORDERPRE(bithptr  *p )
        {
            bithptr  *q;
            if(p->ltag==1)
                ____①____;
            else
            {
                ____②____;
                while ( ____③____ )
                    ____④____;
                ____⑤____;
            }
        }
```

6 以下是在中序线索二叉树中查找结点*p 的中序后继算法，请填空。

```
typedef  enum  PointerTag{Link,Thread};
typedef  struct  BiThrNode
{
    TElemType  Data;
    struct BiThrNode  *Lchild,*Rchild;
    PointerTag  Ltag,Rtag;
}BithrTree;

BithrTree  *INORDERNEXT(BithrTree  *p )
{
    BithrTree  *q;
    if(p->rtag==1)
```

```
    ____①____;
    else
    {
    ____②____;
    while ( _____③_____ )
    ____④____;
    ____⑤____;
    }
}
```

7 下面算法的功能是从带头结点的单链表中查找值为 X 的结点，并将其删除（设这样的结点不多于一个），请将其补充完整。

```
    /*单链表结点结构定义*/
    typedef struct node
    {   int data;
        struct node *next;}linklist;
    /*算法: 从单链表中查找值为 X 的结点，若存在则删除之*/
    DELETEX(linklist *&head, int x)
    {   ____①____ ;  /*定义变量*/
        p=head;q=p->next;
        while (  ____②____  )
        {   p=q;q=q->next;}
        if (  ____③____  )
        { printf( "Node not Found!" );return;}
        else
        {   ____④____;
            ____⑤____;
            printf( "Node has been deleted!" );
            return;
        }
    }
```

三、算法设计题

1 设计算法求一个带头结点的单链表的所有结点值的和。

2 n 个整数存放在一维数组中，0 元素未用，设计顺序查找算法,查找值为 x 的元素是否存在，若存在返回位置，否则返回 0。

3 根据顺序表的定义，设计算法在表中查找值为 x 的数据是否存在，存在返回位置，否则返回-1。

```
#define ListSize 100
typedef struct
{   datatype data[ListSize];
```

```
    int length;
}SqList;
```

4 设计算法求一个带头结点的单链表的所有结点值的和。已知单链表的结点结构类型如下:

```
typedef struct LNode
{  int  data;
struct LNode *next;}LNode;
```

5 根据下面的类型定义，写出顺序栈的入栈算法（8分）。

```
#define  maxstacksize 100
typedef struct
{   dataType base[maxstacksize];
    dataType *Top;
}sqstack;
```

6 根据栈的定义，设计读取栈顶元素的算法（8分）。

```
    typedef struct
    {
        datatype *base;
        datatype *top;
        int stacksize;
    }SqStack;
```

7 一批整数存放在带头结点的单链表中（结点类型为 Lnode），请写出求这些整数中最小值的算法。