一、

1. 算法指的是（　　）。

A 计算机程序　　B 解决问题的计算方法

C 排序方法　　D 解决问题的有限运算序列

2. 在数据的树形结构中，数据元素之间为（　　）的关系。

A　0:0　　B　1:1　　C　1:n　　D　m:n

3. 在一个长度为 n 的线性表中，删除值为 x 的元素时需要比较元素和移动元素的总次数为（　）。

A.　(n+1)/2　　B.　n-1　　C.　n　　D.　n+1

4. 在一个单链表中删除 p 所指向结点的后继结点时，其算法的时间复杂度为（　　）。

A. O(1)　B. O(n/2)　　C. O(n)　D. $O(n^2)$

5. 下列程序段的时间复杂度为（　　）。

```
x=n;  //n>1
y=0;
while (x>=(y+1)*(y+1))
y=y+1;
```

A、O(n)　　B、$O(\sqrt{n})$　　C、O(1)　　D、$O(n^2)$

6. 一棵非空二叉树的先序遍历序列和后序遍历序列正好相反，则该二叉树一定满足（　　）。

A. 所有结点均无右孩子　　B. 所有结点均无左孩子

C. 任意一棵二叉树　　D. 只有 1 个叶子结点

7. 有一个 100 阶的三对角矩阵 M，其元素 $m_{i,j}$(1<=i<=100, 1<=j<=100),按行优先存储方法存到从 0 下标开始的一维数组 N 中，则元素 $m_{30,30}$ 在数组中的下标为（　　）。

A.　89　　B.　88　　C.　87　　D.　86

8.对于 KMP 算法，在模式匹配时指示主串匹配位置的指针（　　）。

A. 不会变大　B. 不会变小　　C. 无法判断　D. 都有可能

9.若长度为 n 的线性表采用顺序存储结构，在其第 i 个位置插入一个新元素算法的时间复杂度为（　　）。

A、$O(\log_2 n)$　　B、O(1)

C、O(n)　　D、$O(n^2)$

10. 在一个单链表中，若要在 P 所指向的结点之后插入一个新结点，则需要相继修改（）个指针域的值.

A. 1　　B.　2　　C. 3　　D. 4

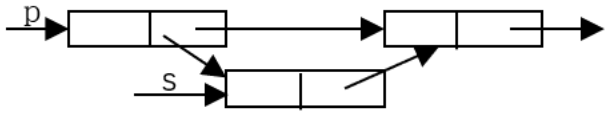
11.将下图所示的 s 所指结点加到 p 所指结点之后，其语句应为（　　）。

A、s－＞next=p+1;p－＞next=s;

B、(*p).next=s;(*s).next=(*p).next;

C、s－＞next=p－＞next;p－＞next=s－＞next;

D、s－＞next=p－＞next;p－＞next=s;



12.在双向链表存储结构中，删除 p 所指的结点时须修改指针（　　）。

A、p－＞next－＞prior=p－＞prior;p－＞prior－＞next=p－＞next;

B、p－＞next=p－＞next－＞next;p－＞next－＞prior=p;

C、p－＞prior－＞next=p;p－＞prior=p－＞prior－＞prior;

D、p－＞prior=p－＞next－＞next;p－＞next=p－＞prior－＞prior;

13.在双向循环链表中，在 P 指针所指的结点后插入 q 所指向的新结点，其修改指针的操作是（　　）。

A、p－＞next=q; q－＞prior=p;p－＞next－＞prior=q;q－＞next=q;

B、p－＞next=q;p－＞next－＞prior=q;q－＞prior=p;q－＞next=p－＞next;

C、q－＞prior=p;q－＞next=p－＞next;

p－＞next－＞prior=q;p－＞next=q;

D、q－＞next=p－＞next;q－＞prior=p;p－＞next=q;p－＞next=q;

14.将两个各有 n 个元素的有序表归并成一个有序表，其最少的比较次数是（　　）。

A、 n　b.2n－1　c.2n　d.n－1

15.在一个长度为 n 的顺序表中，在第 i 个元素（1≤i≤n）之前插入一个新元素时须向后移动（　　）个元素。

A、n－i　B、n－i+1　C、n－i－1　D、i

16. 已知一棵完全二叉树的第 6 层有 8 个叶子结点，则该完全二叉树的结点个数最多是（　　）。

A. 39　　B.52　　C.　111　　D. 119

17. 广义表 L=（(a,b,c,d)）的表头是（　　），表尾是（　）。

A.　a　　B. ( )　　C. （a,b,c,d）　　D.　（b,c,d）

18.若指定有 n 个元素的向量，则建立一个有序单向链表的时间复杂性的量级是（　　）。

A、O(1)　　B、O(n)　　C、$O(n^2)$　　D、$O(n\log_2 n)$

19. 广义表 L=（(a,b,c)）的深度和长度分别为（　）。

A. 3 和 1　　B.2 和 1　　C.　3 和 2　　D.1 和 1

20. 若字符串 S=“software”，则其子串的数目是（　　）。

A. 8　　B. 9　　C.　36　　D. 37

21. 串“ababaaababaa”的 next 数组为（　　）。

A. 012345678999　B.012121111212　　C.011234223456　　D.0123012322345

22．若用一个大小为 6 的数组来实现循环队列，且当 rear 和 front 的值分别为 0 和 3。当从队列中删除一个元素，再加入两个元素后，rear 和 front 的值分别是（　　）。

A、1 和 5　　B、2 和 4

C、4 和 2　　D、5 和 1

23．设栈的输入序列为 1、2、3、4，则（　　）不可能是其出栈序列。

A、1243　　B、2134　　C、1432　　D、4312　　E、3214

24．设栈的输入序列是 1、2、…、n，若输出序列的第一个元素是 n，则第 i 个输出元素是（　　）。

A、不确定　　B、n-i+1　　C、i　　D、n-i

25．假定一个顺序循环队列的队首和队尾指针分别用 front 和 rear 表示，则判队空的条件是（　　）。

A、front+1==rear　　B、front==rear+1

C、front==0　　D、front==rear

26．假定一个顺序循环队列存储于数组 A[n]中，其队首和队尾指针分别用 front 和 rear 表示，则判断队满的条件是（　　）。

A、(rear-1)%n==front　　B、(rear+1)%n==front

C、rear==(front-1)%n　　D、rear==(front+1)%n

27．一个栈的的输入序列为 12345，则下列序列中不可能是栈的输出序列的是（　　）。

A、23415　　B、54132　　C、23145　　D、15432

28.若一个栈的输入序列是 1、2、3、…、n，输出序列的第一个元素是 i，则第 i 个输出元素是（　　）。

A、i-j-1　　B、i-j　　C、j-i+1　　D、不确定

29．用单链表表示的链式队列的队头在链表的（　　）位置。

A、链头　　B、链尾　　C、链中

30．若 X 是二叉中序线索树中一个有左孩子的结点，且 X 不为根，则 X 的前驱为(　　)。

A．X 的双亲　　B．X 的右子树中最左的结点

C．X 的左子树中最右叶结点　　D．　X 的左子树中最右结点

31．在下列算法描述中，涉及到队运算的算法是（　　）。

A、表达式求值算法　　B、深度优先搜索

C、二叉树遍历　　D、广度优先搜索

32.栈的插入和删除操作在（　　）进行。

A、栈顶　　B、栈底　　C、任意位置　　D、指定位置

33．在一个具有 n 个顶点的有向图中，若所有顶点的出度之和为 s，则所有顶点的入度之和为（　　）。

A、s　　B、s-1　　C、s+1　　D、n

34．当利用大小为 N 的数组存储顺序循环队列时，该队列的最大长度为（　　）。

A、N-2　　B、N-1　　C、N　　D、N+1

35．如果以链表作为栈的存储结构，则退栈操作时（　　）。

A、必须判别栈是否满　　B、判别栈元素的类型

C、必须判别栈是否空　　D、对栈不作任何判别

36．链栈与顺序栈相比有一个明显的优点，即（　　）。

A、插入操作更加方便　　B、通常不会出现栈满的情况

C、不会出现栈空的情况　　D、删除操作更加方便

37．设有两个串 p 和 q，求 q 在 p 中首次出现的位置的运算称作（　　）。

A、连接　　B、求子串　　C、模式匹配　　D、求串长

38. 为解决计算机主机与打印机之间速度不匹配问题，通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是（　　）。

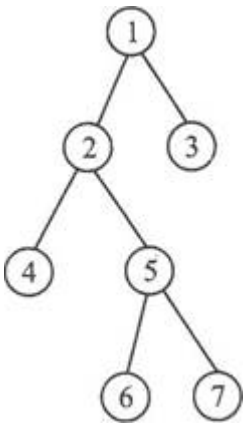
A．栈　　B．队列　　C．树　　D．图

39．设栈 S 和队列 Q 的初始状态均为空，元素 a，b，c，d，e，f，g 依次进入栈 S。若每个元素出栈后立即进入队列 Q，且 7 个元素出队的顺序是 b，d，c，f，e，a，g，则栈 S 的容量至少是（　　）。

A．1　　B．2　　C．3　　D．4

40．　给定二叉树如下图所示。设 D 代表二叉树的根，L 代表根结点的左子树，R 代表根结点的右子树。若遍历后的结点序列为 3，1，7，5，6，2，4，则其遍历方式是（　　）。

A．LRD　B．DRL　　C．RLD　　D．RDL



41．判定一个有向图是否存在回路，除了可以利用拓扑排序的方法外，还可以利用（　　）。

A．求关键路径的方法　　B.求最短路径的 Dijkstra 方法

C．深度优先遍历算法　　D. 广度优先遍历算法

42．就排序算法所用的辅助空间而言，选择排序、快速排序、归并排序的关系是（　　）。

A、选择排序<快速排序<归并排序　　B、选择排序<归并排序<快速排序

C、选择排序>归并排序>快速排序　　D、选择排序>快速排序>归并排序

43．若数据元素序列 11，12，13，7，8，9，23，4，5 是采用下列排序方法之一得到的第二趟排序后的结果，则该排序算法只能是（　　）。

A．起泡排序　B．插入排序　C．选择排序　D．二路归并排序

44．将森林转换为对应的二叉树，若在二叉树中，结点 u 是结点 v 的父结点的父结点，则在原来的森林中，u 和 v 可能具有的关系是（　　）。

I．父子关系 II．兄弟关系 III．u 的父结点与 v 的父结点是兄弟关系

A．只有I　B．I和II　C．I和III　D．I、II和III

45．循环队列的队满条件为 （ ）

A.　(sq->rear+1) % mazsize ==(sq->front+1) % maxsize;　B. (sq->.rear+1 % maxsize ==sq->front+1

C．(sq->(rear+1) % maxsize ==sq->front　　D. sq->rear ==sq->front

46．一棵非空的二叉树的先序遍历序列与后序遍历序列正好相反，则该二叉树一定满足（　　）。

A、所有的结点均无左孩子　　B、所有的结点均无右孩子

C、只有一个叶子结点　　D、是任意一棵二叉树

47．一棵完全二叉树上有 1001 个结点，其中叶子结点的个数是（　　）。

A、250　　B、500　　C、254　　D、505　　E、以上答案都不对

48．对记录的关键字集合 key={50，26，38，80，70，90，8，30，40，20}进行排序，各趟排序结束时的结果为：

50	26	38	80	70	90	8	30	40	20
50	8	30	40	20	90	26	38	80	70
26	8	30	40	20	80	50	38	90	70
8	20	26	30	38	40	50	70	80	90

其使用的排序方法是（　　）。

A、快速排序　　B、基数排序　　C、希尔排序　　D、归并排序

49. 若需在 O(nlog2n)的时间内完成对数组的排序，且要求排序是稳定的，则可选择的排序方法是（　　）。

A、快速排序　　B、堆排序　　C、归并排序　　D、直接插入排序

50.若一棵完全二叉树有 768 个结点，则该二叉树中叶子结点的个数为（　　）。

A. 257　B. 258　C. 384　D. 385

51．任何一棵二叉树的叶结点在前（先）序、中序和后序遍历序列中的相对次序（　　）。

A、不发生变化　　B、发生变化　　C、不能确定

52．设 a、b 为一棵二叉树上的两个结点。在中序遍历时，a 在 b 前面的条件是（　　）。

A、a 在 b 的右方　　B、a 在 b 的左方

C、a 是 b 的祖先　　D、a 是 b 的子孙

53．设有 13 个值，用它们组成一棵哈夫曼树，则该哈夫曼树共有（　　）个结点。

A、13　　B、12　　C、26　　D、25

54．下面几个符号串编码集合中，不是前缀编码的是（　　）。

A、{01，0000，0001，001,1}　　B、{011，000，001，010，1}

C、{000，001，010，011,100}　　D、{0,100,110,1110,1100}

55．某二叉树中序序列为 ABCDEFG，后序序列为 BDCAFGE，则前序序列是（　　）。

A、EGFACDB　　B、EACBDGF　C、EAGCFBD　　D、上面的都不对

56．对二叉排序树进行（　　）遍历，可以得到该二叉树所有结点构成的排序序列。

A、前序　　B、中序　　C、后序　　D、按层次

57. 在对 n 个元素进行冒泡排序的过程中，最好情况下的时间复杂性为（　　）。

A、O(1)　　B、O(nlog2n)　　C、O(n2)　　D、O(n)

58.在一棵深度为 h 的完全二叉树中，所含结点的个数不小于（　　）。

A、2^h　　B、2^{h+1}　C、2^h-1　D、2^{h-1}

59．下列选项给出的是从根结点分别到达两个叶子结点路径上的权值序列，能属于同一棵哈夫曼树的是（　　）。

A. 24,10,5 和 24，10，7　　B. 24，10，5 和 24，12，7

C. 24，10，10 和 24，14，11　　D.24，10，5 和 24，14，6

60．采用散列表存储的目的是（　　）。

A、插入　　B、删除　C、快速查找　D、排序

61．设散列函数为 H(k)=k mod 7,一组关键码为 23，14，9，6，30，12 和 18，散列表 T 的地址空间为 0.6，用线性探测法解决冲突，依次将这组关键码插入 T 中，得到的散列表为（　　）。

A.

0	1	2	3	4	5	6
14	6	23	9	18	30	12

B.

0	1	2	3	4	5	6
14	18	23	9	30	12	6

C.

0	1	2	3	4	5	6
14	12	9	23	30	18	6

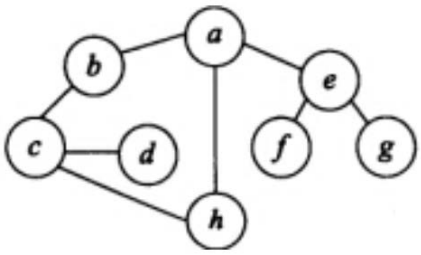
D.

0	1	2	3	4	5	6
6	23	30	14	18	12	9

62．若用邻接矩阵存储有向图，矩阵中主对角线以下的元素均为零，则关于该图拓扑序列的结论是（ ）。

A．存在，且唯一　B．存在，且不唯一　C．存在，可能不唯一　D．无法确定是否存在

63．若对下图所示的无向图进行遍历,则下列选项中, 不是广度优先遍历序列的是(　　)。



A. h,c,a,b,d,e,g,f　　B. e,a,f,g,b,h,c,d　　C. d,b,c,a,h,e,f,g　　D. a,b,c,d,h,e,f,g

64. 对于有 n 个顶点，e 条边且使用邻接表存储的有向图进行广度优先遍历，其算法的时间复杂度为(　　)。

A、O(n+e)　　B、O(n)　　C、O(e)　　D、O(n*e)

65．一个 n 个顶点的连通无向图，其边的个数至少为 （　　）。

A、n-1　　B、n　　C、n+1　　D、$n\log_2 n$

二、

1. 下面程序段的时间复杂度是_____。

```
x=0;
for(i=1;i<n;i++)
for(j=1;j<=n-i;j++)
        x++;
```

2.下面程序段的时间复杂度是_____。

```
int i,j,k;
 for(i=0;i<n;i++)
 for(j=0;j<=n;j++)
    {  c[i][j]=0;
                  for(k=0;k<n;k++)
        c[i][j]=a[i][k]*b[k][j]
     }
```

3. 下面程序段的时间复杂度是_____。

```
i=n-1;
        while((i>=0) && (A[i]!=k))
          j- -;
          return (i);
```

4.下面程序段的时间复杂度是_____。

```
fact(n)
```

```
{ if(n<=1)
    return (1);
  else
    return (n*fact(n-1));
  }
```

5. 下面程序段的时间复杂度是_____。

```
for(i=0;i<n;i++)
for(j=0;j<n;j++)
A[i][j]=0;
```

6. 下面程序段的时间复杂度是_____。

```
i=s=0;
while(s<n)
{  i++;
   s+=i;
}
```

7. 下面程序段的时间复杂度是_____。

```
s=0;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
s+=B[i][j];
sum=s;
```

8. 下面程序段的时间复杂度是_____。

```
i=1;
while(i<=n)
i=i*3;
```

9.设语句 x++的时间是单位时间，则以下语句的时间复杂度为_____。

```
for(i=1; i<=n; i++)
for(j=i; j<=n; j++)
x++;
```

10.下列程序段的时间复杂度为_____。

```
x=n;    /*n>1*/
y=0;
while (x>=(y+1)*(y+1))
```

y=y+1;

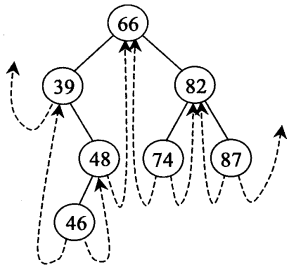
三、

1. 设通信中出现 5 种字符分别是 A, B, C, D, E, 出现的概率分别为 0.2, 0.05, 0.4, 0.1, 0.25，请为它们构造哈夫曼树，并对这些字符完成编码。

2. 已知一棵线索化的二叉排序树如图所示。

（1）说明该树的线索化是基于何种遍历次序的；

（2）在该树中插入元素值为 53 的结点并修改相应线索，画出修改之后的线索二叉排序树。



3. 已知线性表为顺序表存储，阅读以下代码，并回答问题：

（1）设线性表 L＝（21,-7,-8,19 ,0,-11,34,30,-10）含有 9 个元素,写出执行算法 test1 后的 L 状态;

(2) 简述算法 test1 的功能；

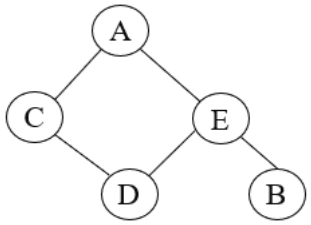
```
void test1(sqlist *L)
{int i, j;
  for(i=j=0; j<L->length; i++)
    if (L->data[i]>=0)
    {if (i!=j) L->data[j]=L->data[i]; j++
```

```
    }
  L->length=j;
  }
```

4. 已知主串为“ccgcgccgcgcgcg”，模式串为“cgcgcg”。下表为按照朴素的模式匹配算法进行的前两趟匹配。请继续完成剩下各趟匹配过程。并说明：（1）匹配成功，共进行了几趟模式匹配？（2）一趟匹配失败时 i, j 的值？（3）匹配成功，算法的返回值？

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
	c	c	g	c	g	c	c	g	c	g	c	g	c	g	
第 1 趟	c	g													匹配失败时 i=2,j=2
第 2 趟		c	g	c	g	c	g								匹配失败时 i=7,j=6

5. 有一个图的边集为{(A,C),(A,E),(B,E),(C,D),(D,E) }，假如按字母正序采用邻接矩阵 存储，回答下列问题： （7 分） （1）请画出该图； （2）从顶点 a 出发进行深度优先搜索遍历得到的顶点 DFS 序列为？ （3）从顶点 a 出发进行广度优先搜索遍历得到的顶点 BFS 序列为？



6. 设散列表为 HT[13], 散列函数为 H (key) = key %13。用闭散列法解决冲突, 对下列关 键码序列 12, 23, 45, 57, 20, 03, 78, 31, 15, 36 构造表。采用线性探查法寻找下 一个空位, 现已有前 4 个关键字构造的散列表如下所示。回答问题： （1）请将剩余 6 个关键字填入表中相应位置； （2）计算等概率下查找成功的平均查找长度是？

0	1	2	3	4	5	6	7	8	9	10	11	12
					57	45				23		12

7. 以下算法是关于二叉排序树的运算，试分析算法的功能。

```
int n=0;
typedef   struct   node
{ int key;
  struct node *lchild,*rchild;
}bitree;
void BST(bitree *t, int x)
{ if(t!=NULL)
   { n++;
     if(bt->key==x) return;
     else if (bt->key>x)
              BST(bt->lchild, x);
          else
              BST(bt->rchild, x);
     }
}
```

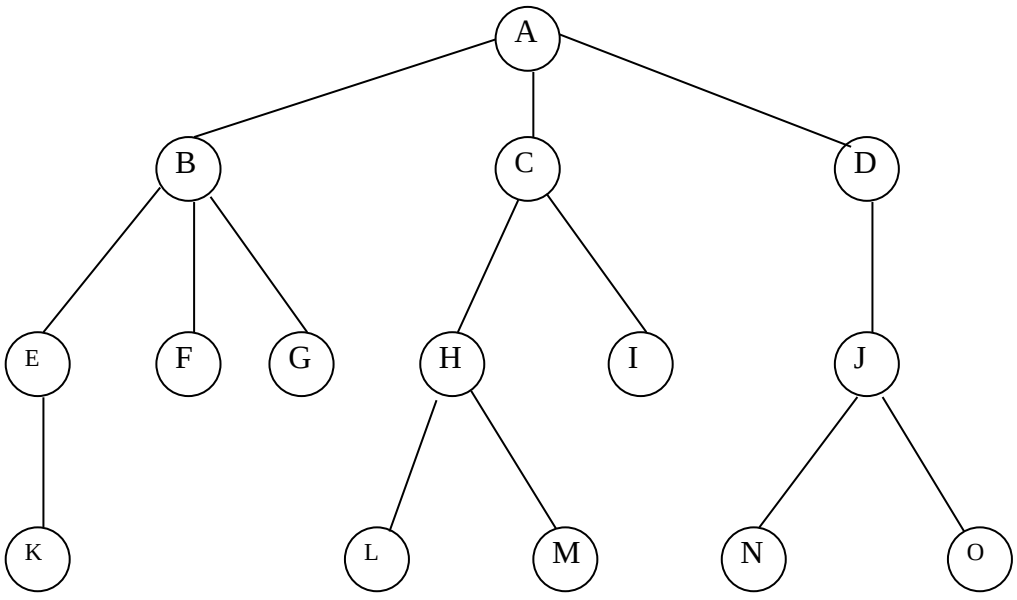
8. 已知有向图的邻接表的类型定义如下：要求编写求有向图 G 中第 i 个顶点（顶点序号 0～n-1）的出度。函数定义为 int outdegree(ALGraph * g, int i) 。

```
#define maxsize 100; //图的最大顶点数
typedef struct node
{ int adjvex; //邻接点域
   struct node *next;
}edgenode;//边表结点类型

typedef struct nodevex
{ char vextex;//顶点域
   edgenode *link;//边表头指针
} vexnode;//顶点表结点类型
typedef struct nodegraph
{ vexnode ga[maxsize];//邻接表
   int n, e;//图中当前的顶点数和边数
} ALGraph;//邻接表类型


    int outdegree(ALGraph * g, int i)
       {
```

}

9. 已知一棵二叉树的前序遍历序列是 ABDGCEFH，其中序遍历序列为 DGBAECHF。请画出相应的二叉树，并求出对应此二叉树的后序遍历序列，此二叉树是完全二叉树吗？完全二叉树有什么性质（特点）？

10、写出下面的树转化为二叉树后的前序、中序和后序遍历序列。



11. 假设以带头结点的单循环链表作为非递减有序线性表的存储结构。请设计一个时间复杂度为 0(n) 的

算法，删除表中所有数值相同的多余元素，并释放结点空间。
例如：（7，10，10，21，30，42，42，42，51，70），经算法操作后变为（7，10，21，30，42，51，70）

链表的结点定义如下：

```
typedef struct node
{ datatype data;
   struct node *next;
}linklist;

void delete(linklist*head)
{

}
```

12. 借助栈（可直接调用栈的基本运算 InitStack( ), Push( ), Pop( )）来实现带有头结点的单链表的逆置算法。

单链表结点定义：

```
typedef struct node
{ datatype data;
   struct node *next;
}Linklist;
```

顺序栈定义：

```
# define maxsize 1024
typedef char datatype;
typedef    struct stack
```

```
{ datatype data[maxsize];
   int Top;
}stack ;

void converse(linkList*head)
   {
```

```
      }
```

13. 设线性表 A=（a_1,a_2,…a_m）,B=(b_1,b_2,…b_n),试写一个按下列规则将线性表 A、B 合并成线性表 C 的算法，使得

    C=（a_1,b_1,…a_m,b_m,b_{m+1},…b_n）当 m<=n 时；

    C=（a_1,b_1,…a_m,a_n,a_{n+1},…a_n）当 m>n 时。

线性表 A、B 和 C 均以带头结点的单链表作为存储结构，且 C 表利用 A 表和 B 表中的结点空间构成，C 表仅头结点可以另开辟空间。请将算法的 4 处补充完整。

```
  typedef struct node
  { int data;//数据域
    struct node *next;//指针域
```

```
}Linklist;

 Linklist *merge(Linklist *A, Linklist *B)
  //A、B 分别指向 A 表、B 表，返回 C 表指针
 { Linklist *q,*p;
  Linklist *C=( Linklist *)malloc(sizeof(Linklist));
    C->next=NULL；
    q=C；
    while(A->next!=NULL&&B->next!=NULL)
      {p= A->next;
       A->next=P->next;
       （1）_____
       （2）_____
       p=B->next;
       B->next=P->next;
       （3）_____
       （4）_____
       }
  if (A->next!=NULL)      q->next=A->next;
  if (B->next!=NULL)      q->next=B->next;
  return C;
 }
```