**《数据结构》上机实验题目**

**（共8次，每次上机4小时）**

**第一阶段（线性部分）**

**《数据结构》第1上机题 （线性表练习）**

1．编程实现书P19 ADT List 基本操作12个：

* 1. 用顺序存储结构实现； （2）用链式存储结构实现；

2．设元素值为整型的线性表L，分别采用顺序结构和链式结构存储，编写程序，实现线性表的就地逆置（习题集P18 2.21， 2.22）。

3．输入正整数n、m（m<n），设有n个人坐成一圈，从第1个人开始循环报数，报到m的人出列，然后再从下一个人开始报数，报到m的人又出列，如此重复，直到所有的人都出列为止。要求用链式结构和顺序结构实现，按出列的先后顺序输出每个人的信息。

4. CSP题目 小明上学

问题描述：小明是附属中学的一名学生，他每天都要骑自行车往返于家和学校。他希望能够预计自己上学所需要的时间。他上学需要经过数段道路，相邻两段道路之间设有至多一盏红绿灯。红绿灯是这样工作的：每盏红绿灯有红、黄、绿三盏灯和一个能够显示倒计时的显示牌。假设红绿灯被设定为红灯 r 秒，黄灯 y 秒，绿灯 g 秒，那么从 0 时刻起，[0,r) 秒内亮红灯，车辆不许通过；[r, r+g) 秒内亮绿灯，车辆允许通过；[r+g, r+g+y) 秒内亮黄灯，车辆不许通过，然后依次循环。倒计时的显示牌上显示的数字 l（l > 0）是指距离下一次信号灯变化的秒数。一次上学的路上，小明记录下了经过每段路的时间，和各个红绿灯在小明到达路口时的颜色和倒计时秒数，计算此次小明上学所用的时间。

输入格式

　　输入的第一行包含空格分隔的三个正整数 r、y、g，表示红绿灯的设置。这三个数均不超过 106。  
　　输入的第二行包含一个正整数 n（n ≤ 100），表示小明总共经过的道路段数和看到的红绿灯数目。  
　　接下来的 n 行，每行包含空格分隔的两个整数 k、t。k=0 表示经过了一段道路，耗时 t 秒，此处 t 不超过 106；　　　　k=1、2、3 时，分别表示看到了一个红灯、黄灯、绿灯，且倒计时显示牌上显示的数字是 t，此处 t 分别不会超过 r、y、g。

输出格式

　　输出一个数字，表示此次小明上学所用的时间。

样例输入

30 3 30  
8  
0 10  
1 5  
0 11  
2 2  
0 6  
0 3  
3 10  
0 3

样例输出

70

样例说明

小明先经过第一段道路，用时 10 秒，然后等待 5 秒的红灯，再经过第二段道路，用时 11 秒，然后等待 2 秒的黄灯和 30 秒的红灯，再经过第三段、第四段道路，分别用时6、3秒，然后通过绿灯，再经过最后一段道路，用时 3 秒。共计 10 + 5 + 11 + 2 + 30 + 6 + 3 + 3=70 秒。

评测用例规模与约定

测试点 1, 2 中不存在任何信号灯。  
　　测试点 3, 4 中所有的信号灯在被观察时均为绿灯。  
　　测试点 5, 6 中所有的信号灯在被观察时均为红灯。  
　　测试点 7, 8 中所有的信号灯在被观察时均为黄灯。  
　　测试点 9, 10 中将出现各种可能的情况。

5. CSP题目 学生排队

问题描述

　　体育老师小明要将自己班上的学生按顺序排队。他首先让学生按学号从小到大的顺序排成一排，学号小的排在前面，然后进行多次调整。一次调整小明可能让一位同学出队，向前或者向后移动一段距离后再插入队列。

　　例如，下面给出了一组移动的例子，例子中学生的人数为8人。

　　0）初始队列中学生的学号依次为1, 2, 3, 4, 5, 6, 7, 8；

　　1）第一次调整，命令为“3号同学向后移动2”，表示3号同学出队，向后移动2名同学的距离，再插入到队列中，新队列中学生的学号依次为1, 2, 4, 5, 3, 6, 7, 8；

　　2）第二次调整，命令为“8号同学向前移动3”，表示8号同学出队，向前移动3名同学的距离，再插入到队列中，新队列中学生的学号依次为1, 2, 4, 5, 8, 3, 6, 7；

　　3）第三次调整，命令为“3号同学向前移动2”，表示3号同学出队，向前移动2名同学的距离，再插入到队列中，新队列中学生的学号依次为1, 2, 4, 3, 5, 8, 6, 7。

　　小明记录了所有调整的过程，请问，最终从前向后所有学生的学号依次是多少？

　　请特别注意，上述移动过程中所涉及的号码指的是学号，而不是在队伍中的位置。在向后移动时，移动的距离不超过对应同学后面的人数，如果向后移动的距离正好等于对应同学后面的人数则该同学会移动到队列的最后面。在向前移动时，移动的距离不超过对应同学前面的人数，如果向前移动的距离正好等于对应同学前面的人数则该同学会移动到队列的最前面。

输入格式

　　输入的第一行包含一个整数n，表示学生的数量，学生的学号由1到n编号。

　　第二行包含一个整数m，表示调整的次数。

　　接下来m行，每行两个整数p, q，如果q为正，表示学号为p的同学向后移动q，如果q为负，表示学号为p的同学向前移动-q。

输出格式

　　输出一行，包含n个整数，相邻两个整数之间由一个空格分隔，表示最终从前向后所有学生的学号。

样例输入

8

3

3 2

8 -3

3 -2

样例输出

1 2 4 3 5 8 6 7

问题分析：这个问题可以通过顺序结构或（双向或单向）链表实现，但对于移动元素较多的情况，应采用哪种存储结构更优呢？

选做题：习题集 1.19 1.20 2.19

**《数据结构》第2上机题 （线性表练习）**

1. 设元素值为整型的线性表L，分别采用顺序结构和链式结构存储，编写程序，用选择排序算法实现线性表的表排序。

2．设线性表A、B，元素值为整型，且递增有序，编写程序，实现下列功能：对采用顺序结构和链式结构2种存储结构，要求在A的空间上构成一个新线性表C，其元素为A和B元素的并集，且表C中的元素值递增有序（互不相同）。

3．CSP题目

**问题描述：**先输入一个十进制整数n，再输入n个正整数，求它们相邻数之差可知是否为上升或下降，由上升转下降或由下降转上升为折点，求折点数。

**问题分析：**如果一个点的值比左右两个都大或都小，则为折点。

样例输入：

5

1 3 5 2 1

样例输出

1

样例输入：

6

3 5 1 7 8 4

样例输出

3

4．CSP题目

问题描述：小H和小W来到了一条街上，两人分开买菜，他们买菜的过程可以描述为，去店里买一些菜然后去旁边的一个广场把菜装上车，两人都要买n种菜，所以也都要装n次车。具体的，对于小H来说有n个不相交的时间段[a1,b1],[a2,b2]…[an,bn]在装车，对于小W来说有n个不相交的时间段[c1,d1],[c2,d2]…[cn,dn]在装车。其中，一个时间段[s, t]表示的是从时刻s到时刻t这段时间，时长为t-s。  
　　由于他们是好朋友，他们都在广场上装车的时候会聊天，他们想知道他们可以聊多长时间。  
输入格式  
　　输入的第一行包含一个正整数n，表示时间段的数量。  
　　接下来n行每行两个数ai，bi，描述小H的各个装车的时间段。  
　　接下来n行每行两个数ci，di，描述小W的各个装车的时间段。  
输出格式  
　　输出一行，一个正整数，表示两人可以聊多长时间。  
样例输入  
4  
1 3  
5 6  
9 13  
14 15  
2 4  
5 7  
10 11  
13 14  
样例输出  
3  
数据规模和约定  
　　对于所有的评测用例，1 ≤ n ≤ 2000, ai < bi < ai+1，ci < di < ci+1,对于所有的i(1 ≤ i ≤ n)有，1 ≤ ai, bi, ci, di ≤ 1000000。  
　　给两个人设定两个数组t[i]，t1[i] 当装车时置1，当t[i]=t1[i]时 总数sum++。

选做题：习题集 2.24 2.29 2.30

**《数据结构》 第3次上机题（线性表复习，栈与队列练习）**

1．编程实现书P45 ADT Stack 基本操作9个，用顺序存储结构实现；

2．编程实现书P59 ADT Queue 基本操作9个，用链式存储结构实现；

3. 利用栈操作实现八皇后问题求解。

**4.** CSP题目

**问题描述：**首先输入正整数n，接着输入n个正整数，如果其中存在一个数，比该数大的个数等于比该数小的个数，则输出该数，如果不存在则输出-1。

**问题分析：**这个问题可以用排序来实现。

样例输入：

6

9 7 2 5 7 8

样例输出

7

样例输入：

5

9 5 2 5 7

样例输出

-1

5. CSP题目

问题描述

　　在某图形操作系统中,有 N 个窗口,每个窗口都是一个两边与坐标轴分别平行的矩形区域。窗口的边界上的点也属于该窗口。窗口之间有层次的区别,在多于一个窗口重叠的区域里,只会显示位于顶层的窗口里的内容。  
　　当你点击屏幕上一个点的时候,你就选择了处于被点击位置的最顶层窗口,并且这个窗口就会被移到所有窗口的最顶层,而剩余的窗口的层次顺序不变。如果你点击的位置不属于任何窗口,则系统会忽略你这次点击。  
　　现在我们希望你写一个程序模拟点击窗口的过程。

输入格式

输入的第一行有两个正整数,即 N 和 M。(1 ≤ N ≤ 10,1 ≤ M ≤ 10)  
　　接下来 N 行按照从最下层到最顶层的顺序给出 N 个窗口的位置。 每行包含四个非负整数 x1, y1, x2, y2,表示该窗口的一对顶点坐标分别为 (x1, y1) 和 (x2, y2)。保证 x1 < x2, y1 < y2。  
　　接下来 M 行每行包含两个非负整数 x, y,表示一次鼠标点击的坐标。  
　　题目中涉及到的所有点和矩形的顶点的 x, y 坐标分别不超过 2559 和　　1439。

**问题分析：**这个问题可以用链式线性表来实现。

输出格式：输出包括 M 行,每一行表示一次鼠标点击的结果。如果该次鼠标点击选择了一个窗口,则输出这个窗口的编号(窗口按照输入中的顺序从 1 编号到 N);如果没有,则输出"IGNORED"(不含双引号)。

样例输入

3 4  
0 0 4 4  
1 1 5 5  
2 2 6 6  
1 1  
0 0  
4 4  
0 5

样例输出

2  
1  
1  
IGNORED

样例说明

　　第一次点击的位置同时属于第 1 和第 2 个窗口,但是由于第 2 个窗口在上面,它被选择并且被置于顶层。  
　　第二次点击的位置只属于第 1 个窗口,因此该次点击选择了此窗口并将其置于顶层。现在的三个窗口的层次关系与初始状态恰好相反了。  
　　第三次点击的位置同时属于三个窗口的范围,但是由于现在第 1 个窗口处于顶层,它被选择。  
　　最后点击的 (0, 5) 不属于任何窗口。

选做题：习题集 2.32 2.37 2.38

**《数据结构》 第4次上机题（线性结构练习）**

1．输入稀疏矩阵，建立稀疏矩阵三元组顺序结构，实现转置（1、2）；

2. 求矩阵的马鞍点。（习题集5.19）

3. CSP题目

问题描述：首先输入正整数n（n<10000），接着输入n个正整数（最大值为10000），对于这n个数，统计输出其中的相邻数对（差值为1的数对），相同数据只被统计一次。

**问题分析：**这个看似是一个ｎ个数与n个数进行比较（O(n2)）的问题,能否用高效的方法解决？

样例输入

6  
1 3 8 2 5 2

样例输出

2  
样例输入

6  
4 3 6 3 5 2

样例输出

4  
4. CSP题目

问题描述：请实现一个铁路购票系统的简单座位分配算法，来处理一节车厢的座位分配。  
　　假设一节车厢有20排、每一排5个座位。为方便起见，我们用1到100来给所有的座位编号，第一排是1到5号，第二排是6到10号，依次类推，第20排是96到100号。  
　　购票时，一个人可能购一张或多张票，最多不超过5张。如果这几张票可以安排在同一排编号相邻的座位，则应该安排在编号最小的相邻座位。否则应该安排在编号最小的几个空座位中（不考虑是否相邻）。  
　　假设初始时车票全部未被购买，现在给了一些购票指令，请你处理这些指令。

输入格式：对于所有评测用例，1 ≤ n ≤ 100，所有购票数量之和不超过100。

　　输入的第一行包含一个整数n，表示购票指令的数量。  
　　第二行包含n个整数，每个整数p在1到5之间，表示要购入的票数，相邻的两个数之间使用一个空格分隔。

输出格式

输出n行，每行对应一条指令的处理结果。  
　　对于购票指令p，输出p张车票的编号，按从小到大排序。

**问题分析：**这个问题可以用顺序结构或链式结构实现。

样例输入

4  
2 5 4 2

样例输出

1 2  
6 7 8 9 10  
11 12 13 14  
3 4

选做题：习题集 3.20 3.28 3.32

**第一阶段总结**

**第二阶段（树与图部分）**

**《数据结构》 第5次上机题目 (二叉树练习 )**

1. 编程实现书P121 ADT BinaryTree 基本操作20个，用二叉链表结构实现；

2.实现二叉树的先序、中序、后序遍历，用递归和非递归方法；实现层次遍历。

3. 编程实现，对二叉树中每个元素值为x的结点，删除以它为根的子树，并释放相应空间。（习题集6.45）

4. 编程实现，判断给定的二叉树是否是完全二叉树。（习题集6.49）

5． CSP题目 消除类游戏

问题描述

　　消除类游戏是深受大众欢迎的一种游戏，游戏在一个包含有n行m列的游戏棋盘上进行，棋盘的每一行每一列的方格上放着一个有颜色的棋子，当一行或一列上有连续三个或更多的相同颜色的棋子时，这些棋子都被消除。当有多处可以被消除时，这些地方的棋子将同时被消除。

　　现在给你一个n行m列的棋盘，棋盘中的每一个方格上有一个棋子，请给出经过一次消除后的棋盘。

　　请注意：一个棋子可能在某一行和某一列同时被消除。

输入格式

　　输入第一行包含两个整数n, m，用空格分隔，分别表示棋盘的行数和列数。

　　接下来n行，每行m个整数，用空格分隔，分别表示每一个方格中的棋子的颜色。颜色使用1至9编号。

输出格式

　　输出n行，每行m个整数，相邻的整数之间使用一个空格分隔，表示经过一次消除后的棋盘。如果一个方格中的棋子被消除，则对应的方格输出0，否则输出棋子的颜色编号。

样例输入

4 5

2 2 3 1 2

3 4 5 1 4

2 3 2 1 3

2 2 2 4 4

样例输出

2 2 3 0 2

3 4 5 0 4

2 3 2 0 3

0 0 0 4 4

样例说明

　　棋盘中第4列的1和第4行的2可以被消除，其他的方格中的棋子均保留。

样例输入

4 5

2 2 3 1 2

3 1 1 1 1

2 3 2 1 3

2 2 3 3 3

样例输出

2 2 3 0 2

3 0 0 0 0

2 3 2 0 3

2 2 0 0 0

样例说明

　　棋盘中所有的1以及最后一行的3可以被同时消除，其他的方格中的棋子均保留。

问题分析：这个问题与树无关，可以使用二维数组来存储，通过一遍遍历对符合条件的格子进行标记，然后第二遍遍历时消除符合条件的格子。

选做题：习题集 6.42 6.45 6.47

**《数据结构》 第6次上机题目 ( 二叉树、树、图练习 )**

1．编程实现书P156 ADT Graph 基本操作13个，用邻接矩阵存储结构实现；

2. 输入N个权值（1-100正整数），建立哈夫曼树。

3. 编程实现，对一棵以孩子-兄弟链表表示的树，输出第i层的所有元素。

4. CSP题目 碰撞的小球

问题描述

　　数轴上有一条长度为L（L为偶数)的线段，左端点在原点，右端点在坐标L处。有n个不计体积的小球在线段上，开始时所有的小球都处在偶数坐标上，速度方向向右，速度大小为1单位长度每秒。

　　当小球到达线段的端点（左端点或右端点）的时候，会立即向相反的方向移动，速度大小仍然为原来大小。

　　当两个小球撞到一起的时候，两个小球会分别向与自己原来移动的方向相反的方向，以原来的速度大小继续移动。

　　现在，告诉你线段的长度L，小球数量n，以及n个小球的初始位置，请你计算t秒之后，各个小球的位置。

提示

　　因为所有小球的初始位置都为偶数，而且线段的长度为偶数，可以证明，不会有三个小球同时相撞，小球到达线段端点以及小球之间的碰撞时刻均为整数。

　　同时也可以证明两个小球发生碰撞的位置一定是整数（但不一定是偶数）。

输入格式

　　输入的第一行包含三个整数n, L, t，用空格分隔，分别表示小球的个数、线段长度和你需要计算t秒之后小球的位置。

　　第二行包含n个整数a1, a2, …, an，用空格分隔，表示初始时刻n个小球的位置。

输出格式

　　输出一行包含n个整数，用空格分隔，第i个整数代表初始时刻位于ai的小球，在t秒之后的位置。

样例输入

3 10 5

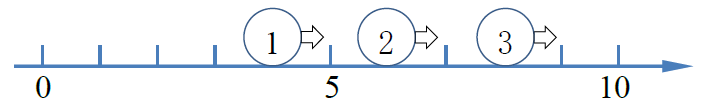
4 6 8

样例输出

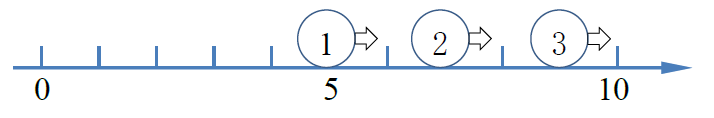
7 9 9

样例说明

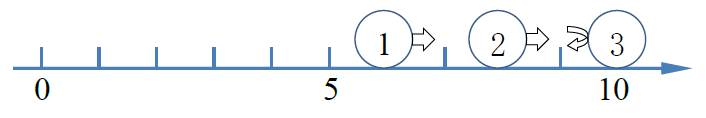
　　初始时，三个小球的位置分别为4, 6, 8。



　　一秒后，三个小球的位置分别为5, 7, 9。



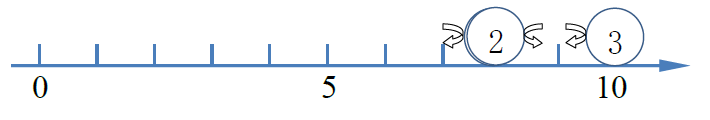
　　两秒后，第三个小球碰到墙壁，速度反向，三个小球位置分别为6, 8, 10。



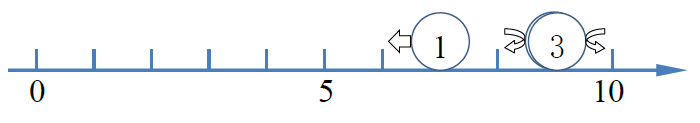
　　三秒后，第二个小球与第三个小球在位置9发生碰撞，速度反向（注意碰撞位置不一定为偶数），三个小球位置分别为7, 9, 9。



　　四秒后，第一个小球与第二个小球在位置8发生碰撞，速度反向，第三个小球碰到墙壁，速度反向，三个小球位置分别为8, 8, 10。



　　五秒后，三个小球的位置分别为7, 9, 9。



样例输入

10 22 30

14 12 16 6 10 2 8 20 18 4

样例输出

6 6 8 2 4 0 4 12 10 2

问题分析：该问题中，只需维护好每个小球的当前位置，并随时间进行变化，最后按题设要求进行输出即可，本题目实质是一道简单的模拟题，与树、图无关。

选做题：习题集 6.49 6.60 6.62

**《数据结构》 第7次上机题目 ( 图 练习 )**

1. 图的深度优先和广度优先遍历；

2．编写Dijkstra算法；

3．CSP题目

问题描述 ：Alice和Bob正在玩井字棋游戏。 井字棋游戏的规则很简单：两人轮流往3\*3的棋盘中放棋子，Alice放的是“X”，Bob放的是“O”，Alice执先。当同一种棋子占据一行、一列或一条对角线的三个格子时，游戏结束，该种棋子的持有者获胜。当棋盘被填满的时候，游戏结束，双方平手。   
　　Alice设计了一种对棋局评分的方法：   
　　- 对于Alice已经获胜的局面，评估得分为(棋盘上的空格子数+1)；   
　　- 对于Bob已经获胜的局面，评估得分为 -(棋盘上的空格子数+1)；   
　　- 对于平局的局面，评估得分为0；

　　例如上图中的局面，Alice已经获胜，同时棋盘上有2个空格，所以局面得分为2+1=3。   
　　由于Alice并不喜欢计算，所以他请教擅长编程的你，如果两人都以最优策略行棋，那么当前局面的最终得分会是多少？   
输入格式   
　　输入的第一行包含一个正整数T，表示数据的组数。   
　　每组数据输入有3行，每行有3个整数，用空格分隔，分别表示棋盘每个格子的状态。0表示格子为空，1表示格子中为“X”，2表示格子中为“O”。保证不会出现其他状态。   
　　保证输入的局面合法。(即保证输入的局面可以通过行棋到达，且保证没有双方同时获胜的情况)   
　　保证输入的局面轮到Alice行棋。   
输出格式   
　　对于每组数据，输出一行一个整数，表示当前局面的得分。   
样例输入   
3   
1 2 1   
2 1 2   
0 0 0   
2 1 1   
0 2 1   
0 0 2   
0 0 0   
0 0 0   
0 0 0   
样例输出   
3   
-4   
0   
样例说明   
　　第一组数据：   
　　Alice将棋子放在左下角(或右下角)后，可以到达问题描述中的局面，得分为3。   
　　3为Alice行棋后能到达的局面中得分的最大值。   
　　第二组数据：

　　Bob已经获胜(如图)，此局面得分为-(3+1)=-4。   
　　第三组数据：   
　　井字棋中若双方都采用最优策略，游戏平局，最终得分为0。

4．CSP题目

问题描述：小刘承包了很多片麦田，为了灌溉这些麦田，小刘在第一个麦田挖了一口很深的水井，所有的麦田都从这口井来引水灌溉。 为了灌溉，小刘需要建立一些水渠，以连接水井和麦田，小刘也可以利用部分麦田作为“中转站”，利用水渠连接不同的麦田，这样只要一片麦田能被灌溉，则与其连接的麦田也能被灌溉。现在小刘知道哪些麦田之间可以建设水渠和建设每个水渠所需要的费用（注意不是所有麦田之间都可以建立水渠）。请问灌溉所有麦田最少需要多少费用来修建水渠。

输入格式：输入的第一行包含两个正整数n, m，分别表示麦田的片数和小刘可以建立的水渠的数量。麦田使用1, 2, 3, ……依次标号。 接下来m行，每行包含三个整数ai, bi, ci，表示第ai片麦田与第bi片麦田之间可以建立一条水渠，所需要的费用为ci。

输出格式：输出一个整数，表示灌溉所有麦田所需要的最小费用，及水渠连接说明。

**问题分析：**这个问题可以用最小生成树算法实现。

输入样例:

4 4

1 2 1

2 3 4

2 4 2

3 4 3

输出样例:

6

建立以下3条水渠：麦田1与麦田2、麦田2与麦田4、麦田4与麦田3。

选做题：习题集 7.27 7.34 7.37

**第二阶段总结**

**第三阶段（查找与排序部分）**

**《数据结构》 第8次上机题目 ( 查找 排序 练习 )**

1. 实现二叉排序树的插入和删除。

2．实现交换、选择、归并等简单排序算法；

3．实现快速排序算法；

4．实现堆排序算法；

5．CSP题目 数字排序

问题描述

　　给定n个整数，请统计出每个整数出现的次数，按出现次数从多到少的顺序输出。

输入格式

　　输入的第一行包含一个整数n，表示给定数字的个数。

　　第二行包含n个整数，相邻的整数之间用一个空格分隔，表示所给定的整数。

输出格式

输出多行，每行包含两个整数，分别表示一个给定的整数和它出现的次数。按出现次数递减的顺序输出。如果两个整数出现的次数一样多，则先输出值较小的，然后输出值较大的。

样例输入

12

5 2 3 3 1 3 4 2 5 2 3 5

样例输出

3 4

2 3

5 3

1 1

4 1

问题分析：该题目可用一个数组，以下标作为数，数组内容存储该数出现次数来实现（这就相当于直接映射，是机试题目里面常用的一种解题法，很多看似非线性的题型最后其实都可以采取哈希或者映射的方法来巧解，尤其是一些看似是树形结构的模拟类题目，要好好体会哈希思想在机试题目中的巧用）。

选做题：习题集 9.32 10.32 10.34

**第三阶段总结**