

- **1002 A+B for Polynomials**

- 注意保留一位小数
- 注意要删去和中系数为0的项，如果结果为0则输出0

- **1005 Spell It Right**

- 基本题意：输入一个整数，计算每一位的加和，将最后的结果的每一位用英语输出，比如加和为15，则输出one five。
- 方法： 10^{100} ，最多一百位，只要一个数组就可以了！！（这方法太巧妙了！！随着输入加和。每一位最大为9，则加和也最大为九百。（nb
- 用递归的方法来输出每位数字

```
void dfs(int n){
    if(n/10 == 0){
        printf("%s", number[n%10]);
        return;
    }
    dfs(n/10);
    printf("%s", number[n%10]);
}
```

- **1006 Sign In and Sign Out**

- 不要一味执着于C++，对于既有字符串又有数字，可以用C来代替
- 善用struct

- **1009 Product of Polynomials**

- 可以边输入第二个多项式边进行计算

- **1010 Radix**

- 主要题意：给两个数，其中一个数已经给了进制，另一个没给，判断这两数有没有可能相等
- 思路
 - 先把给了进制的转换成10进制
 - 然后用二分法来判断有没有可能
 - 这里用到isdigit函数和max_element()函数，string可以rbegin和rend
 - low是字符串中最大的那个数，high是max (low, 已知的十进制 number)
 - 记得用long long 判断是否溢出，即小于0

- **1015 Reversible Primes**

- 这题真的很绕。。大概题意：给一个数和进制数，判断该数是否为素数，并且判断该数在x进制下反过来再转回十进制是不是素数，如果是就yes。
- 举例：23 2->23是素数，满足第一个条件->23转到二进制是10111，反过来是11101->转回十进制是29，29也是素数，over

- **1012 The Best Rank**

- 大概题意：每个人有三个科目，和平均分。需要对每个科目的所有学生的成绩进行排序，每个学生拥有单科成绩排名，最后输出单科最高排名
- 需要注意
 - 解题思路：设置一个结构存放学生的成绩，一个数组存放排名
 - 需要注意的是
 - 排名时相同分数记作排名相同，12245这样
 - 初始化数组时需要在main函数之前初始化，感觉类似map的功能

- **1013 Battle Over Cities**

- 一堆城市，每一个都应该能通到每一个里面去，此时敌军破坏一个城市，其它的城市需要两两可以到达，要求破坏掉一个城市后，需要补的路的数量
- 考察：DFS
- 注意点：
 - 最后一个测试点超时：用scanf!!!
 - 测试点2过不去：题目中给出来的刚开始的不一定是联通的，还可能有单独的孤立的点！他给你的图不一定是一张连通图，甚至还会有孤立顶点，你得把这些图全都连起来。
 - **如果之后有时间最好试一试数组来做**

- **1016 Phone Bills**

- 大致题意：算电话账
- 注意
 - 在记录每个小时里面的分钟数多少钱时，可以多出一个来计算整个一天算作多少钱
 - 计算钱数时，可以从00:00:00开始算，不然需要处理
 - 不同天
 - 同一天不同小时
 - 同一天同意小时不同分钟
 - **很麻烦！**

- **1019 General Palindromic Number**

- 不要**先算再整合成数组**再判断是否是这种数，直接**边计算边整合成数组**之后再判断

- **1029 Median (可以再用two pointers再做一遍)**

- 找到两组数的中间值
- 用set就完事了

- **1032 Sharing**

- 大概题意：找到两个单词中开始相同的点
- 主要思路：
 - 设置一个struct Node，包括每个点代表的data和nextnode和isVisited，存储起来。
 - 首先把第一个单词遍历一遍，把其中的isVisited设置成true。
 - 再遍历第二个单词，当遍历带到的字母在第一个单词中已经isVisited了，就

直接输出这个点的地址，否则输出-1。

- **1033 To Fill or Not to Fill 贪婪问题 好不友好**

- 问题简述：从一个地方到另一个地方，中间有不同价格的加油站，需要计算出最少的钱
- 解题思路：
 - 1.除了把所有该录进去的录进去，还要在最后添加一个重点距离最大且price为0的station
 - 2.如果第一个距离不是0直接break掉
 - 3.主体部分
 - 从目前所在的下一个开始找，一直到现在的距离+最远行驶距离处的加油站，判断其中的最小price（此时如果所有的price都大于现在的就找到最小的，如果找到第一个小于当前price的就赶紧break），以此来得到最小价格和该价格对应的station
 - 找到之后，算出需要多少
 - ◆ 如果最小价格小于现在的价格
 - ◇ 需要的油量大于现有的油量，加到需要的油量，更新走到下一个站点时的油量=0
 - ◇ 需要的油量小于现有的油量，直接走到下一个站点，更新油量
 - ◆ 否则，直接加满就好，然后更新走到下一个站点时的油量

- **1035 Password**

- 基本题意：看是否有需要修改的密码，修改一下
- 注意当不需要修改密码的时候，如果N数量为1和数量为其它时不一样

- **1038 Recover the Smallest Number**

- 拼接字符串，搞出最小数字
- 注意：
 - 在拼接时进行排序，排序的标准为判断 $s1+s2 < s2+s1$ ，由此来得到从小到大的顺序
 - 当整个字符串的长度不为0并且第一位仍然是0时，erase掉，最后判断如果erase完了之后size为0则输出0，否则输出字符串

- **1039 Course List for Student**

- 大概题意就是看哪个学生上了哪几门课

- **1046 Shortest Distance**

- 注意运行超时问题，解决方案为在输入数据的时候就加上（数组中每一个数字都是从1到该位置的距离），两个距离一个相减得到，一个由sum-第一个距离得到

- **1052 Linked List Sorting**

- 就是输入一个链表，然后给链表里的数排序的题
- 坑：
 - 当链表里啥也没有的时候，要输出0 -1

- **1053 Path of Equal Weight**

- 题意：从根节点到叶子节点，给定某个值，所有的weight加起来等于这个值，输

出所有的weight

- 思路就是：
 - 首先输入，边输入边给某个节点的所有孩子排序
 - 存一个vector的temp，一个以vector为单位的vector的ans
 - 进行DFS，遇到条件合适的就把temp push进ans
 - 测试点1是遇到s=root的weight时，记得输出

- **1055 The World's Richest**

- 基本题意：输出某个年龄段的富人
- 两个超时问题
 - 先进行排序
 - 排序后筛掉每个年龄里面>100个的那些人，将所有年龄的在一百个之内的人存入一个新的数组
 - (! 不要在结构中定义数组!! 注意!!)

- **1060 Are They Equal**

- 佩服自己奇奇怪怪的思路.....多看看自己写的屎山吧55

- **1061 Dating**

- 注意：HH和MM都要注意0~9的情况，01，02这种
- 可以看一下书上：先读入四个字符串再比较，将星期几当作二维char类型数组来保存，到时候再找。

- **1062 Talent and Virtue**

- 大概题意：根据德和才两个指标排序
- 注意
 - 不需要弄四个数组，直接在结构体中添加一个flag标志就好了
 - 另外：输入字符串还是用char数组比较好

- **1063 Set Similarity**

- 主要题意：看两个set 的相似度
- 注意：用vector-set存放数据，不要二次insert，不然会超时

- **1065 A+B and C (64bit)**

- 注意正溢出和负溢出，正溢出即两个正数相加得负数，负溢出即两个负数相加得正数。
- long long a,b,c
- 一定要记得sum也一定要是long long

- **1067 Sort with Swap(0, i)**

- 基本题意：每次只能交换数组中的两个数字，计算最少需要多少次能交换完
- 注意：
 - 每次都交换位置0的数字和0所在的位置
 - 分为两种情况
 - 位置0上就是0：往后面找第一位不是0的和0交换
 - 位置0上不是0：交换位置0的数字和0所在的位置

- **1069 The Black Hole of Numbers**

- 输入数字，从大到小排序-从小到大排序 = 结果
- 注意：

- 输入的时候不一定是四位数字，还有可能是百位十位这种，12，145这种
- 需要输入字符串，在前面insert 0就可以了，用到的函数是
`s.insert(index,count,ch)`，表示在index位置插入count个字符ch
- **1070 Mooncake**
 - 每种月饼的库存要用double存，我也不知道为啥(
- **1073 Scientific Notation**
 - 注意，在指数为正数时，指数大小是否比小数点的位数多，比如
 - 1.23456×10^2 和 1.23456×10^{10} 是不一样的
- **1075 PAT Judge**
 - 题意：给PAT的排分数
 - 注意：当编译不通过时需要判断此时的分数，如果分数为-1/0则赋值为0，如果前面已经有通过了的，那就不用赋值为0了
- **1076 Forwards on Weibo**
 - 找到几层之内转发了这条微博的人数
 - 注意点：用BFS！因为用DFS会超时55555
- **1078 Hashing**
 - Quadratic probing
 - 这个的意思是 $(\text{number} \% \text{table_size}) + 1^2$ ， $(\text{number} \% \text{table_size}) - 1^2$ ， $(\text{number} \% \text{table_size}) + 2^2$ ， $(\text{number} \% \text{table_size}) - 2^2$题目中的意思是只需要 $+1^2$ ， $+2^2$ 这样
 - 有的数字是一直找不到地方的，所以+到 $(\text{table_size}-1)^2$ 就可以了，具体解释可以看算法笔记的笔记
- **1080 Graduate Admission**
 - 基本题意：学校录取学生，学校有招生名额，学生有志愿和分数
 - 注意
 - 在后面排序之后，i不再是stu.id！！
 - 此外就是 一定要注意注意注意如果上来这个学校就是一个也不录取的情况！
- **1085 Perfect Sequence**
 - 找perfect子序列，用二分法
 - 注意： $m \times p$ 的时候可能会超出int范围，用long long int
- **1089 Insert or Merge**
 - 给出两段序列，判断从第一段到第二段经历了插入排序还是归并排序
 - 解题方法
 - 首先找到b中最后一个按从大到小排的数i，然后从i+1开始赋值于j，然后找和a中不同的数
 - 如果j==n说明是插入排序，直接对a,a+i+2进行排序就可以了
 - 如果否，则说明是归并排序，这个就一步步来，从 $a+i \times k$, $a+(i+1) \times k$ 进行归并排序，将 $k=k \times 2$ ，直到得到了b，再归并一次就可以了
 - 记得要
 - `sort(a + n / k * k, a + n);`
 - 这里很疑惑的有个浮点错误，原因竟然是找i的时候的 \leq
 - ```
for (i = 0; i < n - 1 && b[i] <= b[i + 1]; i++)
 ;
```

- **1091 Acute Stroke**
  - 主要题意：找到肿瘤块大于t的所有肿瘤块的大小
  - 分析
    - 因为这是一个三维数组，所以要三维分析
    - 其实是一个套路，用BFS做，先把他们输入，然后设置一个isVisited数组，当其中为1并且没被访问过的时候对其进行广度搜索
    - 进入广度搜索后需要加一定的判断条件，来判断数组是否越界以及数组该位置的数字是否满足要求以及是否已经访问过了，这样层层遍历就可以了
  - **有空自己再重新做一遍**
- **1093 Count PAT's**
  - 简单的排列组合问题
  - 唯一需要注意的是，计算A和T的时候都需要%1000000007。
- **1095 Cars on Campus**
  - 基本题意：一堆车，有进入和出去的时间和状态，最终计算在某个时间还在里面的车，和停车最长的时间
  - 解法及注意的点：
    - 需要结构体：①存放所有数据②存放有效数据
    - 需要map来存放某辆车的停留时间
    - 存放有效数据的判断方式：将所有数据按照id排序，次顺序是时间，如果第i条数据与第i+1条数据的id相同，并且i是in，i+1是out的话，就存入valid结构体，此时亦可以计算停留时间（如果有两次in+out一定要都算上）
    - 对于valid结构体中的数据进行按时间排序，然后根据in和out的状态对count进行加减，这样就可以得到每个事件有几辆车辣！
- **1096 Consecutive Factors**
  - 主要题意：找到一个数的连续几个因子乘积（仍然可以被这个数整除）。
  - 方法
    - 设置一个beginnumber，一个length。
    - 从2到sqrt (n) 遍历，设置一个temp用来表示目前乘积，j表示当前下标 = i
    - 用while循环，寻找从i开始的连续几个数字，如果当前length大于原来的，就代替一下，beginnumber为i
    - 之后判断，如果length为0，就是自己本身，其他的就按照顺序排列就好了
- **1097 Deduplication on a Linked List**
  - 将一个链表分成两部分，第一部分是每个跟每个的绝对值都不一样，第二个是剩下的一堆
  - 注意点：
    - 还是要注意第一部分和第二部分的sum1和sum2为0的情况
- **1098 Insertion or Heap Sort**
  - 主要题意：判断是插入排序还是堆排序
  - 注意：
    - 判断插入排序的方法和上面一样，可以用while来判断，更省地
- **1100 Mars Numbers**
  - 地球数字和火星文互换

- 地球转为火星文注意四种情况
  - 0
  - 十位不为0, 个位为0
  - 十位为0, 个位不为0
  - 十位个位都不为0
- 火星文转成地球
  - 需要判别到底十位个位有没有
  - 如果只有一个单词, 判断是十位还是个位
  - 如果有俩单词就顺着来
- 此外: 注意字符串的读入, 用getline
- **1101 Quick Sort**
  - 主要题意: 看哪一个是pivot
  - 思路: 看数组中每一个数字左侧最大值是不是小于该值and右侧最小值是不是大于该值, 如果是, 那么这个就是pivot
  - 注意:
    - 不用实时检测出左侧最大和右侧最小
    - 创建left\_max数组和right\_min数组, 输入完成后再判断出数组中每一个位置处的左边最大和右边最小存入各自数组, 然后再去判断思路中所提到的
- **1104 Sum of Number Segments**
  - 算出矩阵中所有连续片段的和
  - 奇奇怪怪的点
    - 输入数组的每个值的时候要用cin
    - result用 long long然后最后再转换回去
    - 不能用先求和再计算的方式, 输入一个算一个, 每个数字出现的次数为 $i * (n + 1 - i)$
- **1108 Finding Average**
  - 主要题意: 找到一堆数里面合法的数, 给他们求平均值
  - 需要注意的点
    - 12.这种数字是合法的
    - 需要判断的是最后的final\_number的绝对值是否大于1000, 别想着提前省事
- **1117 Eddington Number**
  - 很有意思的一道题, n天骑车大于n公里
  - 这一步很关键
 

```
while(e<n&&a[e]>e+1) e++;
printf("%d",e);
```
- **1118 Birds in Forest**
  - 题意: 找属于同一棵树的鸟
  - 注意点:
    - 注意, 如下面所示情况, 只能有一棵树
      - 1 2 3
      - 4 5 6
      - 2 5 7
    - 多看看柳神的代码 神之所以被成为神



- **1121 Damn Single**

- 找到在宴会上单身的人
- 注意一个坑：输出的时候用%05d

- **1124 Raffle for Weibo Followers**

- 用map

- **1129 Recommendation System**

- 大致题意：随着用户的浏览实时更新推荐的物品清单
- 思路：因为有key有value，需要根据value对key进行排序，而map并没有自带的排序功能，所以可以弄一个struct类的node，将node作为一种数据类型存放成set，node中包含构造函数和<重载函数，之后对于set进行读取，每次读取完后清除掉之前的node，将value+1后insert进去。
- 天，真的是绝了这个思路

```
struct node{
 int index,num;
 node(int a,int b):index(a),num(b){}
 bool operator < (const node &a) const{
 return (num!=a.num) ? num>a.num : index<a.index;
 }
};
```

- **1130 Infix Expression**

- 基本题意：输出一个数学表达式，带括号的！
- 注意：
  - 其实这道题不用建树，用不着那么麻烦，不过没时间改了，凑合这样吧
  - 直接把它用数组/vector存起来，然后直接inorder输出就行，如果是-1就不输出，就完事了
  - 输出括号的问题：
    - 可以看出来，出了最中间的那个root，其它每一个中间的运算符的外围都有括号扩这，所以只需要判断它是个运算符就可以输出（），而判断的条件就是：这个不是根节点，且不是叶子节点
    - 第二个注意点：不是叶子节点的条件：左右有一个不为空即可

- **1133 Splitting A Linked List**

- 分三段重新排列数组，<0的，>=0<=k的，>k的
- 注意点：在result数组中，不要想当然的以为result和list的长度为n，一定要插入几个算几个！用sum来代替真正排列起来到list和result里面的个数！

- **1136 A Delayed Palindrome**

- 也就是继续找加起来的回文
- 一个题干中没有说的测试点
  - 注意，如果一开始给出来的数本身就是一个回文，直接输出就好了，不用reverse再相加

- **1140 Look-and-say Sequence**

- D, D1, D111, D113, D11231, D112213111, ...
- 注意思路

- **1141 PAT Ranking of Institutions**

- 按学校进行PAT 的考试成绩排名



- 注意最后一个测试点是score精度问题，先double最后int过来
- **1145 Hashing - Average Search Time**
  - 其实和之前做的一道题很相似，只是多了一步查询平均次数的计算
  - 思路：
    - 查询平均次数，从 $0^2$ 到 $table\_size^2$ ，加起来最多要查询 $(table\_size+1)$ 次，break的条件是：hash后的数组中对应值为要查询的数，**或者hash后数组中对应位置为0**，这个为0是用来判断根本没有插入进去过的，因为如果要插入早就插入了，为0说明数组中根本没有这个值
- **1149 Dangerous Goods Packaging**
  - 题意：给一堆不能共存的货物，然后再给几个列表，判断这几个列表里面的货能不能呆在一起
  - 注意：
    - 用map记录后面的会有莫名其妙的错误
    - 这么做：
      - 先用map记录下来不能共存的pair货物
      - 然后用vector记录下来每批货物有啥
      - 用a数组记录下来哪个货物是有的（记得初始化）
      - 然后遍历vector，看看v[i]对应的map的那几个不能共存的是不是a里的1，如果是就flag为1就ok
- **1153 Decode Registration Card of PAT**
  - 输入字符串和数字，根据不同的要求输出不同的东西
  - 解题思路：
    - 只需要建立一个node结构，因为情况1 3输出的格式都是字符串+数字
    - 不用把原始字符串拆开用，直接substr就可以
    - 注意substr后一个数字是字符串长度