我自己在准备考研时曾做了下 06，07，08，09 年的题目，并且在博客中提供了一个参考的题解，10 年的题目以及 11 年保研的题目有空我会再做做，毕竟现在上研了没多少时间研究这些，祝大家考出个好成绩，支持王道论坛(www.cskaoyan.com)交大版。

解答请关注我的个人博客。这个 PDF 文档里的参考答案不是我给出的，有些代码是是有问题的，但思路是正确的。

我可能近期内会把历年题目做一个分类，并全部提供一个参考的题解和程序，哪位获得较高初试分数正在备战复试的同学可以与我联系，binlong@sjtu.edu.cn

另外本次重新整理这些题目的时候，基本上都加了一些个人的注解和思路，如果有不对或欠妥的地方或者你有更好的想法，请你与我联系，谢谢！

靖难（http://www.longbin.org）2011 年 2 月 18

2005 年上机试题

问题 1：
网上回忆版的原文如下

太恐怖了，

12 翻一下是 21 对吗？

34 翻一下是 43 对吗？

12+34 是 46 对吗？46 翻一下是 64 对吗？

现在给你 21 与 43，把 64 输出就可以了。

我猜题目的大概意思是给定整数 a,b，若 reverse(a)+reverse(b)==reverse(a+b)则输出 a+b,否则输出 NO

总之这应该是一道颇为简单的题目，鉴于交大喜欢使用 NOIP 原题作为复试题，类似的题目作为练习我推荐一道 NOIP 题目
http://www.rqnoj.cn/Problem_148.html

问题 2：

给你一串路径，譬如

a\b\c

a\d\e

b\cst

d\

你把这些路径中蕴涵的目录结构给画出来，子目录直接列在父目录下面，并比父目录向右缩一格，就象这样

a		
	b	
		c
	d	
		E
b		
	cst	
d		

同一级的需要按字母顺序排列，不能乱。

为了方便看清题目意思，我划表格表达了一下层次关系。本题其实关键是找到合适的数据结构来存储

初步思考了一下，可以考虑使用有序的 N 叉树来存储，第一层目录的你节点记为 0，然后使用类似于先序遍历的方式来遍历，用一个全局变量来记录当前递归深度，并控制空格数量。

有空我会试写一下代码。

问题 3.

回忆版原文如下：

这题听说.....有点问题。反正大概意思是这样的(除非我理解错了.....)：

有一个 x[6][6] 任意的 0<=i,j<=5 1<=x[i][j]<=10;

现在有一个起始位置 i1,j1 与一个结束位置 i2,j2。

请找出一条从 i1,j1 到 i2,j2 的总代价最小的路径。

1. 只能沿上下左右四个方向移动
2. 总代价是每走一步的代价之和
3. 每步（从 a,b 到 c,d)的代价是 x[c][d]与其在 a,b 处状态的乘积
4. 初始状态(在 i1,j1 时的状态）是 1，每走一步，状态按如下公式变化

   (走这步的代价 % 4)+1

   也就是状态只有 4 种:1,2,3 or 4.

这个问题类似于棋盘遍历或者迷宫问题，可以使用 DFS（回溯法）求解，基本的伪代码如下：
全局参数说明：
使用二维数组 G 保存棋盘，与回忆版中的 x 功能相同
变量 x,y 分别表示当前位置
St_x,St_y 表示起始位置
Des_x,Des_y 表示终点位置
二维数组 go[0..3,0..1]中 1..4 表示四个方向，0 表示某个方向时 x 的变化值，1 表示往某个方向时 y 的变化值
比如 go[0][0]表示向右走时 x 的增量
Min 记录当前最小代价，二维数组 path 用于记录当前最小代价路径
Sum_cost 用于求总代价和

代码如下，使用类 pascal：

procedure dfs(k:integer);	
var i:integer;	
begin	
if (x=Des_x) and (y=Des_y) then	如果到达目的地
begin	
if sum_cost<min then	如果当前路径代价总和更小则更新之
begin	
min:=sum_cost;	
for i:=0 to k-1 do	保存当前最小代价的路径
begin	
path[i][0]:=a[i][0];	
path[i][1]:=a[i][1];	
end;	
end;	
sum_cost:=0;	总代价置 0
exit;	
end;	
for i:=0 to 3 do	4 个方向可以尝试
begin	如果这个方向可以走，则尝试一下
if (0=<x+go[i][0]<=5)and(0=<y+go[i][1]<=5)then	
begin	计算这一步的代价
cost:=g[x+go[i][0]][y+go[i][1]]*state[x][y];	计算 总代价
sum_cost:=sum_cost+cost;	更新当前横坐标
x:=x+go[i][0];	更新当前纵坐标
y:=y+go[i][1];	更新当前状态
state[x][y]:=cost mod 4+1;	保存本步坐标
a[k][0]:=x;	
a[k][1]:=y;	回溯找后面的路径
dfs(k+1);	恢复状态，为了不影响后面的其它尝试
sum_cost:=sum_cost-cost;	
x:=x-go[i][0];	
y:=y-go[i][1];	
end;	
end;	
end;	

（当然，本题也可以用 DP 来解）

推荐练习的题目：
可以找相关资料练习相关的经典问题：
全排列问题，整数划分问题，八皇后问题，迷宫问题，马的遍历，0-1 背包问题。
都使用 DFS 来解。

2006 年上机试题

（参考程序是原来网上有人提供的，我只提供了个人的注解和思路）

Problem A.Fibonacci

Input: **fib.in**

Output: **Standard Output**

Time limit: **5 second**

Memory limit: 64 megabytes

The Fibonacci Numbers{0,1,1,2,3,5,8,13,21,34,55...} are defined by the recurre

nce:

F0=0 F1=1 Fn=Fn-1+Fn-2,n>=2

Write a program to calculate the Fibonacci Numbers.

Input

The input file contains a number n and you are expected to calculate Fn.(0<=n<

=30)

Output

Print a number Fn on a separate line,which means the nth Fibonacci Number.

Example

fib.in	Standard Output
1	1
2	1
3	2
4	3
5	5
6	8

太简单，我就不讲了，不过尽量用迭代，不要用递归。

可以试着做一下这道：http://poj.org/problem?id=3070

问题 2：
POJ 原题：http://poj.org/problem?id=2538
网上一堆题解，我就不说了。

问题 3：

Problem C.String Matching

Input: matching.in

Output: Standard Output

Time limit: 5 second

Memory limit: 64 megabytes

Finding all occurrences of a pattern in a text is a problem that arises freque

ntly in text-editing programs.

Typically,the text is a document being edited,and the pattern searched for is

a particular word supplied by the user.

We assume that the text is an array T[1..n] of length n and that the pattern i

s an array P[1..m] of length m<=n.We further assume that the elements of P and

T are all alphabets(∑={a,b...,z}).The character arrays P and T are often cal

led strings of characters.

We say that pattern P occurs with shift s in the text T if 0<=s<=n and T[s+1..

s+m] = P[1..m](that is if T[s+j]=P[j],for 1<=j<=m).

If P occurs with shift s in T,then we call s a valid shift;otherwise,we call s

a invalid shift.

Your task is to calculate the number of vald shifts for the given text T and p

attern P.

Input

In the input file,there are two strings T and P on a line,separated by a singl

e space.You may assume both the length of T and P will not exceed 10^6.

Output

You should output a number on a separate line,which indicates the number of va

lid shifts for the given text T and pattern P.

Example

matching.in	Standard Output
aaaaaa a	6
abababab abab	3
abcdabc abdc	0

网上提供的算法是使用 find 函数，这个得多亏 STL，不过建议学习一下文本匹配的算法，有 DP 的解法。

问题 4：

本题是 NOIP1998 年提高组的题目，BTW~~~~

原题在：

http://www.ntnoi.cn:8080/acmhome/problemdetail.do;jsessionid=21D226999870795A6D771D4529430BD7?&method=showdetail&id=1129

或者

http://oj.jzxx.net/showproblem?problem_id=1571

自己写上去，然后提交吧，或者可以自己参考 NOIP1998 的标程。

英文题目如下：

Input: form.in

Output: Standard Output

Time limit: 5 second

Memory limit: 64 megabytes

Every positive number can be presented by the exponential form.For example,

137 = 2^7 + 2^3 + 2^0

Let's present a^b by the form a(b).Then 137 is presented by 2(7)+2(3)+2(0).

Since 7 = 2^2 + 2 + 2^0 and 3 = 2 + 2^0 , 137 is finally presented by 2(2(2)+2 +2(0))+2(2+2(0))+2(0).

Given a positive number n,your task is to present n with the exponential form which only contains the digits 0 and 2.

Input

The input file contains a positive integer n (n<=20000).

Output

You should output the exponential form of n an a single line.Note that,there s hould not be any additional white spaces in the line.

Example

form.in

137

Stardard Output

2(2(2)+2+2(0))+2(2+2(0))+2(0)

form.in

1315

Stardard Output

2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)

以下是网上提供的 2006 年参考程序：

```
//A.cpp
//18 lines
#include "iostream"
#include "fstream"
using namespace std;

int fib(int n)
{
	if (n==0) return 0;
	else if (n==1) return 1;
	else return fib(n-1)+fib(n-2);
}

void main()
{
	fstream f("fib.in");
	int n;
	f>>n;
	cout<<fib(n)<<endl;
}
```

```
//B.cpp
//28 lines
#include "iostream"
#include "fstream"
#include "string"
#include "stdio.h"
```

```
using namespace std;

char table[100]="`1234567890-=QWERTYUIOP[]\\ASDFGHJKL;'ZXCVBNM,./";

void main()
{
	fstream f("wertyu.in");
	char c[1000];
	string s;
	int tablelen = strlen(table);
	while (!f.getline(c,1000).eof())
	{
		s = c;
		int n = s.length();
		for (int i=0;i<n;i++)
		{
			for (int j=0;j<tablelen;j++)
			{
				if (s[i]==table[j]) s[i]=table[j-1];
			}
		}
		cout<<s<<endl;
	}
}

//C.cpp
//24 lines
#include "iostream"
#include "fstream"
```

```
#include "string"
using namespace std;

int match(string s,string t)
{
	int count=0;
	string::size_type index = -1;
	while ((index = s.find(t,index+1))!=string::npos) count++;
	return count;
}

void main()
{
	fstream f("matching.in");
	string s,t;
	while (!f.eof())
	{
		f>>s;
		f>>t;
		cout<<match(s,t)<<endl;
	}
}



//D.cpp
//60 lines
#include "iostream"
#include "fstream"
#include "string"
using namespace std;
```

```
int getminex(int n)
{
	int k=0;
	int l=1;
	while (1)
	{
		if (n<l) break;
		else
		{
			l=l*2;
			k++;
		}
	}
	return k-1;
}

int getremain(int n)
{
	int t=getminex(n);
	int s=1;
	for (int i=0;i<t;i++)
	{
		s=s*2;
	}
	return n-s;
}

string getform(int n)
{
```

```
        if (n==0) return "";
        else if (n==1) return "2(0)";
        else if (n==2) return "2";
        else if (n==4) return "2(2)";
        else
        {
                string t,s;

                int e = getminex(n);
                if (e == 1) t = "";
                else t = "("+getform(e)+")";
                s = "2"+t;

                int r = getremain(n);
                if (r != 0) s = s+"+"+getform(r);
                return s;
        }
}

void main()
{
        fstream f("form.in");
        int n;
        f>>n;
        cout<<getform(n)<<endl;
}
```

保送生试题:（不知道是哪一年的，而且只有一题的）

Problem D . Code the Tree

本题是 POJ 上的原题：http://poj.org/problem?id=2567

另外有一个推荐的题目：http://poj.org/problem?id=2568

相关题解自己搜索吧，网上一堆。

2007 年复试上机真题：

Problem A. Old Bill

本题其实就用简单的穷举就行了，题目来源 ZOJ：

http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=2679

写好最好放 OJ 上提交一下，防止 BUG 吧。另外 ZOJ 的题解网上也有很多。

题目 B 源自 UVA 上的一个经典的高精度运算的题目，很简单。
http://acm.uva.es/archive/nuevoportal/data/problem.php?p=2781

题目 C 源自 POJ，http://poj.org/problem?id=1775

题目 D
本题是 ACM/ICPC 东北欧区域赛的一道练习赛的题
http://olymp.mephi.ru/statements/problems_ICPC_2010-2011_regional.pdf
题目 Z 便是本题，本题的主要难点在于数据规模。暂时没有想到效率高的解法，以后再更新吧。

Problem A. Prime Number

Input file: *Standard Input*
Output file: *Standard Output*

Time Limit: 1 Second

Output the k-th prime number.

Input

k≤10000

Output

The k-th prime number.

Sample input and output

Standard Input	*Standard Output*
3	5
7	17

Problem B. Simple Sorting

Input file: *Standard Input*
Output file: *Standard Output*

Time Limit: 1 Second

You are given an unsorted array of integer numbers. Your task is to sort this array and kill possible duplicated elements occurring in it.

Input

The first line of the input contains an integer number N representing the quantity of numbers in this array(1≤N≤1000). Next N lines contain N integer numbers(one number per each line) of the original array.

Output

Output file should contain at most N numbers sorted in ascending order. Every number in the output file should occur only once.

Sample input and output

Standard Input	*Standard Output*
6 8 8 7 3 7 7	3 7 8

Problem C. Coincidence

Input file: *Standard Input*
Output file: *Standard Output*

Time Limit: 1 Second

Common subsequence of two string s_1 and s_2 is a pair of sequences of indices $(\{a_i\},\{b_i\})$ such that $a_1 < a_2 < \ldots < a_k$, $b_1 < b_2 < \ldots < b_k$, and $s_1[a_i] = s_2[b_i]$ for all $1 \le i \le k$.
Find a longest common subsequence of two strings.

Input

First and second line of an input contain two strings of lowercase character a…z. There are no spaces before, inside or after the strings. Lengths of strings do not exceed 100.

Output

In the first line of output file k – the length of a longest common subsequence.

Sample input and output

Standard Input	*Standard Output*
abcd cxbydz	2

Problem D. Day of Week

Input file: *Standard Input*
Output file: *Standard Output*

Time Limit: 1 Second

We now use the Gregorian style of dating in Russia. The leap years are years with number divisible by 4 but not divisible by 100, or divisible by 400.
For example, years 2004, 2180 and 2400 are leap. Years 2004, 2181 and 2300 are not leap.
Your task is to write a program which will compute the day of week corresponding to a given date in the nearest past or in the future using today's agreement about dating.

Input

There is one single line contains the day number d, month name M and year number y(1000≤y≤3000). The month name is the corresponding English name starting from the capital letter.

Output

Output a single line with the English name of the day of week corresponding to the date, starting from the capital letter. All other letters must be in lower case.

Sample input and output

Standard Input	*Standard Output*
9 October 2001	Tuesday
14 October 2001	Sunday

Month and Week name in Input/Output:
January, February, March, April, May, June, July, August, September, October, November, December
Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday

Problem E. Pre-Post

Input file: *Standard Input*
Output file: *Standard Output*

Time Limit: 1 Second

We are all familiar with pre-order, in-order and post-order traversals of binary trees. A common problem in data structure classes is to find the pre-order traversal of a binary tree when given the in-order and post-order traversals. Alternatively, you can find the post-order traversal when given the in-order and pre-order. However, in general you cannot determine the in-order traversal of a tree when given its pre-order and post-order traversals. Consider the four binary three below:

All of these trees have the same pre-order and post-order traversals. This phenomenon is not restricted to binary tree, but holds for general m-ary trees as well.

Input

There is only one line of the form m s_1 s_2 indicating that the trees are m-ary trees, s_1 is the pre-order traversal and s_2 is the post-order traversal. All traversal strings will consist of lowercase alphabetic characters. For all input instances, 1≤m≤20 and the length of s_1 and s_2 will be between 1 and 26 inclusive. If the length of s_1 is k(which is the same as the length of s_2, of course), the first k letters of the alphabet will be used in strings.

Output

You should output one line containing the number of possible trees which would result in the pre-order and post-order traversals for the instances. Output value will be within the range of a 32-bit unsigned integer. You are guaranteed that there is at least one tree with the given pre-order and post-order traversals.

Sample input and output

Standard Input
2 abc cba
Standard Output
4

Standard Input
2 abc bca
Standard Output
1

Standard Input
10 abc bca
Standard Output
45

Standard Input
13 abejkcfghid jkebfghicda
Standard Output
207352860

09 年回忆版 I

下午的机试从一点半钟开始，持续到五点。和以前一样，一共 4 道题目。
另外附加一道题目用于试机练习。
上机环境为 MS Visual C++ 6.0，测试是采用黑盒测试的。

四个题目都是英文版的，题目内容大致是这样：

A.输入任意两个日期（年月日，中间不分隔），求这两个日期之间的天数。好像比求任意一天是星期几要简单。

注：这道题其实和 08 年的那道解题思路可以一致。

B.这道题目有点难度，给了一个非常奇怪的函数，让用 C++实现。我按照提示一步一步做了，可就是没得正确出结果来。毅然决定先放弃本题，等后面两题做完再说。

注：这一题题目目前都不知道是什么。。。。

C.输入一个数据矩阵，然后选择这个矩阵中的某些数据，让找出这些数据的最中间那一个。我用的是二路归并排序法解决了这个题目，但是一不小心险些留下一个大 BUG，后来检查的时候及时消除了这个大 BUG。

D.输入一个大文本文档，让判定其中从 A 到 Z 的字母个数。这道题目没有说明输入数据的结束标志，因此我特意问了监考老师，并且得知了他们测试程序的方法——输入输出重定向法，即在命令行状态下输入类似“A.exe < A.txt”的命令测试程序。原来如此，我只需要逐行读取这个文件，直到 EOF 就可以了。因此我用了 while(cin>>A)的循环结构。当我把第四题做完的时候已经是四点一刻了，我回过头来检查了已经做好的这三个题目，保证滴水不漏。然后我又回过头来做第二题，但是最后仍然没得出正确结果。不过我仍然比较高兴，毕竟剩下三个题目基本能保证通过所有黑盒测试。

09 年回忆版 II

今年是 4 题。
1.求二个日期间的天数相差，如果二天是连续的，我们认为有二天。如果数据不符合要求，输出 0；格式是 YYYYDDMM
INPUT：
20090411
20090411

2009.。。(忘记）
2008.。。

2009 2 29
2008 2 29
OUT
1

…

0
这题说白了就是把去年第 4 题给翻新一下，基本思想都不变，注意数据的合法性判断，（超多，我写了一个很长的 JUDGE 函数），这题做起来要点时间，
我花了 30+分钟。
二：
怨念啊怨念，。。。。。intergral 竟然是积分的意思，英文超长，如果有机会看到 PDF 大家就能领略了，给了你一个矩阵，让你求矩阵的积分，。。。
这题超级无语，做了一个半小时，最后实在忍不住问监考学长，学长充满爱怜的问：你是不是不知道什么意思，然后放低声音（就是积分啊）。。。。
周围的同学也一片哗然，估计也没看懂。。。。，立马明白怎么做 但是此时离时间到还有不到 2 分钟，也没心情改了，此处为另一次失误，因为我程序的框架已经完全写好，只要加下求给出数据的面积就行了。而且关键是，到时间了不收卷，竟然在分发奖学金申明要我们签字，此时有同学在狂调，不知道是否能多弄几分。

三：
就是一个模拟题，二个数字，输入后给 4 个数字 a,b,c,d 第一个数组的第 a 到第 b 个，第二个的第 c 到第 d 个。
放到一个新数组，求处于中间位置的数。
就是一个简单的模拟题，估计所有人都能拿分，但是细节还是要注意的。
四：
随便给一行字，要你求出大小写字符出现的次数，
大家估计只要熟悉 getline 和字符 ASCII 码使用的估计都能写出来吧。
也挺简单的 。
我个人做了一 三 四，一个小时出头全部搞定，然后傻看第二题，简直无语，如果今年死在这个题上，也是和交大无缘吧。

10 年回忆版

上午八点半到十一点机试，实际上九点才开始，前半个小时是试机用的，两个半小时解决 4 道题，是比较紧张的。

第一题，后缀数组，唯一一道菜鸟题

输入 grain
对其子串
grain
rain
ain
in
n
分别编号为 0，1，2，3，4
然后对各子串按字典顺序排序，即：
ain,grain,in,n,rain

输出 2，0，3，4，1

大概的思路就是定义一个结构体,包括两个成员，一个是编号，一个是指向该子串位置的指针，然后按字符串比较进行快速排序，输出便可以了。

第二题 最短路径

第一行输入 N,M
N 表示城市的个数，各城市编号为 0 到 N-1，N 的范围我记不清了，不过主要的难题不在这
M 表示道路的条数，道路也是 0 到 M-1 编号，第 K 条道路的长度为 2^K ，问题主要在于 K 可以非常大，比如 495
接下来 M 行输入 M 条道路中每条连结的两个城市编号

输出编号为 0 的城市与其它各城市的最短距离大小，数值太大的以 MOD 100000 的结果输出

这个问题可以直接 Dijkstra 算法，我只做出了 int64 表达范围的结果，超出的要么用大数运算，要么用某些数论的知识。曾经学过有 2 的高次幂 MOD 运算，不过不记得了。

提示：
把存下来的距离改成道路标号就可以了，只存当前到这个节点的每条路径的最大的一个标号，可以证明，如果一条路径的所有标号的最大值大于另一条路径，那么这条路径的代价一定大于另一条，别忘了 2^0+2^1+...2^n = 2^(n+1) - 1 这个基本等式。

第三题 中缀表达式运算

这题其实非常常见，而且没有括号，只有+—*/四种运算，不过也只解决了 60%的数据

注：因为没有括号，所以其实直接扫描就可以了，而且题的数据说明中有说纯+，-的占多少，纯*/的占多少，所以即使你不会处理优先级，也要扫描法把这部分数据搞定。
建议大家把后缀前缀中缀的求值都学会吧，呵呵。

第四题 最小面积子矩阵

定义矩阵中元素个数为矩阵面积
输入 N M K
然后就输入 N 行 M 列的数，在这个矩阵中所有元素和不小于 K 的最小的子矩阵。

这题一看就知道用 DP,可惜一直没想出 DP 的转移方程，就用穷举了，但数据规模太大，穷举也只能过其中部分数据，大约 30%吧

注：暴力的方法比较容易想到，也就是用 DFS~解答树的建立可以参考一下二维的最大子段和问题。不过暴力的解法显然是不能得满分的，数据规模超级大的说。
另外也有 DP 的解法，只不过我懒，就没想咯~有想出来的吱一声。