

两分钟 10 道题，很简单，但是马上进入看题的状态有点麻烦，做了 9 道题。现在只记得第二题是单词可否 组成一个合理的句子，我个人感觉可以。还有一个是几个四边形还有一个五边还是六边形，问你那个图形不一样。第九题只记到好长，不想读，我就直接跳到第十题，然后检查了两三道题。

MIIS

新语言学习题：这部分我没准备，做得有些慢，都是选择题，做到后面有些题拿不准，估计挂在这块了。这门语言叫 MIIS，
[http://en.wikipedia.org/wiki/MIIS_\(programming_language\)](http://en.wikipedia.org/wiki/MIIS_(programming_language))可以提前看看，考察的大都是 arithmetic operation 和 logic operation。

4. 学校 MSII(貌似) 这个新语言，然后回答问题。
介绍你变量类型， 数学计算 $+$ $-$ $*$ $/$ min max
表达式 $>$ $<$ $=$

Math

1. 翻译过来是说 1.5 个人 1.5 天吃 1.5 个 pizza. 问 12 个人 6 天吃多少个 pizza. 一开始楞是没读懂, 花了挺长时间. 英文特别绕. 所以还是注意一下
2. 算时间的一题, 具体描述不记得了, 大家凑活看,
现在 4 点, 20 分钟前是什么 2 点到某点的时间的 3 倍, 问那个时刻是多少分钟前. 也是语言绕.

600

career cup 上面也有部分题目

2,8,18, , 78 问空填什么, : 3 8 18 38 78

1 - 2 有 500cash 和 xxx (数字不记得了) 存款, 用 cash 购买 150 张 0.44 刀的邮票, 2 台 2025 的电脑, 用 cash 支付十五分之一, 存款支付十分之一,
用存款支付 4 名员工一周工资, 四名员工年工资 xxxx 刀。问 cash 剩下多少, 存款剩下多少。

4. 我没看懂, 什么 3: 00pm 5: 00pm 四倍 (求补全)

5. 找规律 3 24 -168 1008 -5040 ?

分别为 8, 7, 6, 5

? 我觉得应该是 20160 正负号可能是 (-1) 的 8 次方, 7 次方之类的

7. 找规律 4 7 13 ? 49 97: 25

一个人买了一种牛 X 的种子种树, 每天树的高度翻倍, 第 10 天使 40m, 第几天树 5m。:7

两个人 1 人办公室在 1, 1 人办公室在 46, 1 人 1 秒走过 5 个办公室, 1 人 1 秒走过 10 个办公室, 在那个办公室门口相遇: 16

两个硬币总和 55, 1 个不是 nickel, 问这两个硬币是什么。(cent, nickel, dime, quarter, dollar): 50 cent&nickel

山羊爬 70.5m 的悬崖, 每天爬 3m 滑 2 米, 问第几天爬顶。:69

apple 40 美分, banana 60, grapefruit 80, 问 pear:40 vowel。

1 个东西 1 美元, 1 人说我买 600, 付了 3 美元, 问买了什么 A six and two zeros

4 点那道题读死我了, 好像是 50 分钟前是两点过后的 4 倍, 问是几点?

几句话里只有一句话是对的判断那句话是对的, 第一句是全队, 第二句好像是下面都不对, 第三话不记得了, 第四第五句是以下不对和全不对, 只记得正确答案是倒数第二个选项。

Coding

coding 1. eval expression, only +, -, *, without (). cal from left to right. (其他面经貌似没有, 新题)

$5 + 8 * 2 = 26$. 用 stack, 把数放进去, 遇到 operator pop 出, 然后运算, 再 push

coding 2. common char, replace most char by least char. 数组, 存所有的 ascii 的 count

1. 54 张牌, 每张牌上 1-9, A-E 各一个数字 + 一个 Letter, 比如 3D, 4F, 5A, 7C; 有一个 string 输入, 写一个 program, 每次读 4 张牌, 1 到 4, 如果发现 1 和 4 letter 一样, 丢弃中间 2 张, 在重新读 1 号位置到 4 号 4 张牌, 如果 1 和 4, number 一样, 4 张全部不要, 如果又没有 number 一样, 也没有 letter 一样, 读 2 到 5 号新的 4 张牌, 再比较 first card 和 last card, 做同样的事情, 最后如果全部扔掉了, print You Win, 如果手里还剩牌, print You lose 和 牌的个数;

object Node(letter, number), string to arraylist 反向,

1. spiral matrix 不过是从右上角开始的 逆时针输出
<http://www.careercup.com/question?id=17118665>
spiral matrix I leetcode
找规律, 终止条件: $addedrow \geq \min(m, n)$

1.

Colorful Numbers. 2345 can be break down to 2, 3, 4, 5, 23, 24, 25, 234, 345。 the products of each substring have no overlapping: 2, 3, 4, 5, 6(2*3), 8(2*4), 10(2*5), 24(2*3*4), 60(2*3*5)。 Give you a number, write a function to determine if it is a colorful number or not.

DFS ,HashSet<Integer>, add(x) == false

```
int index = 0;
list = empty list
while(num != 0)
{
    int r = num%10;
    num = num/10;
    length = list.length;
    for i=index to length
    {
        list.add(list[i] * r);
    }
    list.add(r);
    index = length;
}

hashset set =new hashset;

for i=0 to list.length
{

    set.contains(list[i])
    return false;
    else set.add(list[i]);

return true;
}
```

2. A string 'aBIY' is said to be well ordered because the letters of the string occur one after the other in the alphabet. Write a function where the number of letters in the string are passes as parameter and all such well ordered strings are found. career cup 上有 <http://www.careercup.com/question?id=6206517812396032>

找出所有 well-ordered 的 string, dfs

大小写 "ABcd".toLowerCase()

1. TheHanking Hybrids Company trying to display a vehicle's average miles per gallon of fuel (MPG) and miles per hour (MPH). The on-board computer periodically samples and **records the car's current fuel level (gallons)**. The miles traveled since the last sample, the amount of time (minutes) since the last sample, and the car's current speed (MPH). It also takes samples whenever the vehicle is turned on or off. It saves the data in four arrays. The first element of each array is the trip's first "on" event. "On" events always have a 0 in the minutes array.
2. An example of three arrays:

Fuel	5	4	3	8	7	6	5	4	5	4
Miles	0	40	30	0	70	60	70	30	0	50
Minutes	0	80	45	0	80	80	80	35	0	80
Speed	0	15	0	0	65	55	60	0	0	25
	on		off	on				off	on	

3. This trip average 50 MPG and 43.75 MPH. $350/7$, $350/(480/60)$
 4. Write a function that, given such a array, validates the array and calculate the overall average miles per hour and miles per gallon of fuel for the duration of the trip. You may assume that the cars id never moved between turning it off and turning it back on
1. 需要通过第四行的 speed 来判断开车的时间片，然后算 MPH 和 MPG。
 2. 给你一个 String，让你用 **iterative** 的方法删去重复的字符。例如，一开始是 "1004440056778"，会变成 "10000568"，最后变成 "1568"，返回。

hashmap+stringbuilder。

4.Snake 问题, career cup 上有。<http://www.careercup.com/question?id=14946605>

第一题 是求出字母频率 并且按照出现顺序打印, 还是挺简单的 先用 hashmap 统计频率 再用 hashset 确定打印按照出现顺序: array 字母频率, hashmap<Integer, Character>, hm.keySet()

```
ArrayList sortedKeys=new ArrayList(yourMap.keySet());
Collections.sort(sortedKeys);
```

第二题 经典的 keypad 就是给你对应的表 让你返回相应的字符串, 思路很简单 就是把这个表给赋值到 hashmap 里面。就是赋值起来很繁琐。

第三题 估计就是跪在这道题上面, 给你一个二维 integer board, 让你找出所有最长路, 条件是差的绝对值为 1, 当时已经凌晨, 脑子没反应过来用的是 dfs 现在刚睡醒 突然发现 dp 就能做, 估计跪了 心好塞!!

给一个 string 有 {}, [], () 和字母, 判断是不是 balanced

比如 {a[(b)]} 就是 balanced {a[(]b)} 这就不是 balanced 其实就是判断左边的符号有没有对应到右边的符号。: 我用 stack 做的

2. merge two sorted array, 要去掉有重复的 element。

两个 pointer 扫描就行了

3. "Mingo" 游戏 100*100 的 matrix, 存 1-10000 不同的数字, 往 matrix 放数字, 如果有任意一行或者一列或者对角线

都被访问了, 就出现了 Mingo, 当出现 Mingo 时候提示一下打印出此时放了多少数字了。

Mingo, careecup 上可以搜得到。http://www.glassdoor.com/Interview/The-Second-is-the-Mingo-problem-100-100-board-with-unique-number-from-1-1000000-Each-time-a-number-in-1-1000000-is-called-QTN_239134.htm

Keep a counter for each of the columns and rows. Each time a cell from that row or column is called decrease the counter by one. When any of the counters reach zero call Bingo

不用 extraarray 的方法：每加值到 i, j, 查看下 rowI, colj

```
public class CheckMingle {
    int[][] matrix = new int[10][10];
    static int count = 0;
    int[] row = 10;
    int[] col = 10;
    int diag = 10;
    int diag2 = 10;
    void initial() {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                matrix[i][j] = 0;
            }
        }
    }
    void printMingle() {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[0].length; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
    void isMingo(int[][] matrix) {
        int value = 0;
        int row = 0;
        int col = 0;
        int data = 0;
        for (int i = 0; i < 100; i++) {
            value = (int) (Math.random() * 100);
            row = value / 10;
            col = value - row * 10;
            while (matrix[row][col] != 0) {
                value = (int) (Math.random() * 100);
                row = value / 10;
                col = value - row * 10;
            }
            data = (int) (Math.random() * 1000);
            matrix[row][col] = data;
            checkMingle(matrix, row, col);
        }
    }
    void checkMingle(int[][] matrix, int row, int col) {
        row[row]--;
        col[col]--;
        if(diag) diag1--, diag2--;
    }
}
```

```

        check row, col ,diag1, diag2
    }
    public static void main(String[] args) {
        CheckMingle cm = new CheckMingle();
        cm.initial();
        cm.isMingo(cm.matrix);
        cm.printMingle();
    }
}

```

第一个是说你要给一堆东西打上从 1 到 N 的标签，你有一个打字机。但是打字机的某一个键坏掉了不能用。让你 print 可以打出的最大的标签。比如 N 是 56781212，但是 2 坏掉了，你能打出的最大的标签就是 56781199。（题里的 example 是这样，虽然我觉得如果从 1 连续打的话那么到 2 就跪了。。。。）

当为 0 时，先去前面的 0；若没 0，直接返回

若还有 0，注意 0 用 recursion,5000600 借位 4999999;11023, 10999, 09999,9999;
211023,210999,209999,199999; 1023; 0999; 999

注意结果去 0

类似 next permutation leetcode

第二个是说给你一个矩阵，如果矩阵某个元素比与他相邻的 8 个元素都要大，那么管这个元素叫 mountain。打印出所有的 mountain。Brute force

Count and say leetcode

(4) additive number :

Additive numbers are defined to be a positive integer whose digits form an additive sequence. E.g. 11235 (1+1=2, 1+2=3, 2+3=5). What makes it difficult is that 12,122,436 is also one (12+12=24, 12+24=36). Given a range of integers, find all the additive numbers in that range.

。它麻烦的地方就在于有时候前两个数是 1 位的比如 112xxxx(1+1=2)，但是看到你看到 111 并不能说它不是 additive，因为它可能是 111122xxx(11+11=22)，我后来一算发现如果是对 int 的话，最多就是需要处理三位的情况，因为如果前两个都是四位数，第三个也是四位数那已经要 12 位了，int 没那么大。所以分三种情况判断就行了。

1 位 112,2 位 111122, 三位 111111222

1121315,1112113115

```
public static boolean isAdditiveNumber(String number)
{
    for(int i=1;i<number.length();i++)
    {
        for(int j=i+1;j<number.length();j++)
        {
            int part1_tmp = Integer.parseInt(number.substring(0,i));
            int part2 = Integer.parseInt(number.substring(i,j));
            int index= j;
            int rest = Integer.parseInt(number.substring(j,number.length()));
            while(part1_tmp+part2<=rest)
            {
                int part3 = part1_tmp+part2;
                String str = (new Integer(part3)).toString();
                int length = str.length();
                if(index + length > number.length())
                {
                    break;
                }
                if(number.substring(index,index+length).equals(str))
                {
                    index=length+index;
                    if(index==number.length())
                    {
                        return true;
                    }
                    part1_tmp = part2;
                    part2 = part3;
                }
            }
        }
    }
}
```



```

        rest =
Integer.parseInt(number.substring(index,number.length()));
    }
    else
    {
        break;
    }
}
}
}
return false;
}
}

```

2. 字母变换：一个字符串，一个正整数 n ，字符串中元音字母换成该字母之后的第 n 个元音字母，比如 $\{a,e,i,o,u\}$ 对 $a, n=1$ 和 $n=6$ ，都变成 e 辅音字母换成该字母之前的第 n 个辅音字母。

1. String Mangler。给你一个字符串 S ，把所有元音字母变成大写，所有不是 word 最后一个字符的辅音字母变小写，所有 word 的最后一个字母变大写。

4. Jumpers。类似于跳棋，给你一个棋盘和一个位置，让你计算从这个位置起的最长跳跃距离。每次跳只能越过对方的棋子（限一次）。DFS

$N*N$ 棋盘，0 表示没棋子，1 表示 player1 的棋子，2 表示 player2 的棋子，棋子的坐标必须有且只有一个是偶数，一个棋子可以沿对角方向跳两格当且仅当对角方向第一个位置是对手的棋，第二个位置没有棋，在跳的同时把对手的棋子吃掉。已知棋盘和一个坐标，求该坐标上的棋子能连续跳的最大步数。

1.

4) 按要求生成电话号码。

电话号码每一位都不一样。

一部分数字被屏蔽掉，譬如 1,2 不能出现在号码里面。

所有包含 4 的号码，4 需出现在第一位。

我觉得这题就是 permutation 变换下。

输入包括了屏蔽的数字集 $List<Integer>$ ，需要输出的电话号码长度 $leng$

譬如：

输入 [0,1,2,3,5,6], leng = 3

输出 [[7,8,9],[7,9,8],[8,7,9],[8,9,7],[9,7,8],[9,8,7],[4,7,8],[4,8,7],[4,7,9],[4,9,7],[4,8,9],[4,9,8]]

还是类似与 careercup 上的密码？电话号码？组合题，具体题目记不住了但是类似于：

/*

Print all valid phone numbers of length n subject to following constraints:

1.If a number contains a 4, it should start with 4

2.No two consecutive digits can be same

3.Three digits (e.g. 7,2,9) will be entirely disallowed, take as input

*/

4、Phone Number: 电话号码须遵循这样的规则：（1）、相邻数字不能重复，如 1226 则不行，但 1262 可以。（2）、如果号码中有 4，则该号码必须以 4 开头。请输出所有合法的电话号码。

DFS, 用 list length 来确定是第几个数字

第一题很基础，就是输入一组 int[]，求删去最大三个数之后的平均值。

One pass: l1,l2,l3, sum

If (x > l1) l2 = l1, l3 = l2, l1 = x

Else If (x > l2) l3 = l2, l2 = x

Else If (x > l3) l3 = x

(sum - l1 - l2 - l3)/n

第二题是输入一个字符串比如 gdbAgUvWxyZiK, 忽略大小写，让你求最长的连续字母串，比如这里就是 uvwxyz，返回长度 6。

Cur = 0

While(cur < length) { //string starts with cur

Count = 1

For (I = cur + 1; I < length; i++)

 If (a[i] == a[i-1] + 1) ;count++

 Else break

Cur = i

}

第三题就是经典的打字员的那题，给定每个手指 0-9 对应字母。比如 0--null, 1--a,s,e, 2--g,h, 3-b...然后给你一个输入比如 2301，打出所以可能输出。gba,gbs,gbe,hba...。dfs

(2) letter consecution: input 一个 string，返回所有的连续 substring。连续的 substring 代表字母顺序连续并且三个或以上连续。比如"AbC"就是一个连续的，这个要先处理一下，要么全是 capital 要么全不是，然后遍历一遍就行了。Brute force

(4) tic tac toe: 两个人下五子棋，把最终战局作为 input，是个 boolean 的 array（二维 Boolean 数组，用于表示 board），写程序判断谁胜谁负。规则是连续三个或以上就可以加分，三个连续的加一分，有三个以上连续的加分规则是：n-2。我把行，列和对角线都扫描一遍。然后把最终得分多的作为胜利者。

用两人的分数差表示结果，每行每列只扫描一次

4. programming: 四道题都不是很困难，只能说运气比较好

(2) Long subtraction:

subtraction between two arrays. E.g.

$[2,2,2,2,2] - [1,1,1] = [2,2,1,1,1]$

先把两个 array 补 0 成相同长度

$[2,2] - [0,1]$

$[0,1] - [2,2] = [-2,-1]$

编程题：编程题准备好好说下，一共四题

1. 你有无限多个 3 磅，7 磅，16 磅的砝码，找到最短的组合达到目标 X 磅，如果没有这样的组合，返回 0。

于是想到了 DFS，但是这又不是找到所有的结果，所以找到一个结果就需要返回。于是想到用一个全局 flag 来控制。解决。

3333777

7771616

rec(target, poundIndex, pathLength) { from poundIndex to pounds.length}

1. Anagrams。给你一个字符串 S，找到他的所有 anagram，要求所有非小写字母 at the same positio

找到所有的小写字母，列出所有的 permutation，然后在相应位置插入所有非小写字母。Dfs

一个 list 存插入位置，一个 list 存字母（含重复）

Dfs：

Available letters = 存字母的 list

tempResult

For(letter in availableletters)

Availableletters.remove(letter)

Left = lastIndex+1

Right = thisIndex

tempResult2 += substring(left, right) + letter

Rec(availableletters, letterIndex, tempResult2, resultList)

Availableletters.add(letter)

1. 给一个 $N \times N$ 的 grid，里面有字母，用户输入一个单词，判断这个单词在不在里面，单词可横可竖，不能弯折，如果在，给出所有解。

wordSearch, leetcode

4. 四道编程题：

（1）有个游戏，名叫猜单词。提问者，想一个单词，答题者猜一个单词，两个单词的字数一样。然后提问者要返回 bulls 数和 cows 数。所谓的 bulls 是两个单词中对应位置处字母相同的出现次数。cows 是刨去 bulls 的情况，猜的单词中字母在被猜单词中出现的次数。

[size=14.2857141494751px] [size=14.2857141494751px] 写一个程序，输入两个单词，返回 bulls 数和 cows 数。

[size=14.2857141494751px] [size=14.2857141494751px] 例子：

[size=14.2857141494751px] [size=14.2857141494751px] 单词 猜测 bulls cows

[size=14.2857141494751px] [size=14.2857141494751px] Dusts Studs 1 4

[size=14.2857141494751px] [size=14.2857141494751px] Tracks Mitts 1 1

[size=14.2857141494751px] [size=14.2857141494751px] stats turns 1 1

用 hashmap 存被猜单词的字母(不计重复)

先把 bull 的字母删掉，hm.remove(bullChar)

2. 一个图片，找最小分割数。（一维 DP）

给一个 $m \times n$ 的 2D matrix，1 代表黑字，0 代表白字 假设其为图片，现在需要打印它。n 恰好是纸宽，但一张纸长度只有 1000 像素，所以需要写 function 返回一个 array，即需要在 m 坐标下哪个 index 截断图片。截断那行不能有黑字，如果有黑字，就在它上一行截断。

```
robot, 1+min(x+1,x+1000)
```

m 被分成小于1000的段

dfs, 用dp优化hashmap<分割点, 最小分割数>

rec(curM分割点, 最小分割数)

isvalid(m)

5. Coding

1) 给你一个文本，将 j 行转成列，标点符号和空格都要转
感觉上就是将矩阵顺时针选择 90 度。

譬如：

输入 String[]

```
Today is a sunny day,  
friends come to library,  
play dota!!!
```

```
output [j][n-1-i] = input[i][j];
```

应该输出：

```
p f T  
l r o  
a i d  
y e a  
    n y  
d d  
o s i  
t s  
a c  
! o a  
    m s  
! e u  
! n  
    t n  
    o y
```

l d
i a
b y
r ,
a
r
y
,

2 : greatest possible sum: 给你几个数 eg : 5, 0, -123, 19, -1 找出 greatest possible sum
5+0+19+ (-1) =23 这个 sum 要在 subsequence 里面找, subsequence 是至少有两个数。//这个题我不大明白, 我觉得最大和就是所有负数都不要。但是给的例子我又没有分析明白。求高手

指导。Maximum Subarray, leetcode

- 1、Edge pixel : 用一维数组存放二维图片的 pixel 信息, 每个 pixel 在二维图片中的邻节点 (上, 下, 左, 右, 上左角, 上右角, 下左角, 下右角) 的差值中的最大值与给定的 threshold 的值相比较, 如果 difference 比 threshold 大, 则这个 pixel 是 edge pixel。程序实现找出给定的图中所有的 edge pixel

先写一个 1 维坐标和二维坐标相互转换的 function

```
result.add(getLeft(x))  
result.add(getLeftUp(x))  
collections.max(result) > thershold
```

3.给一个数组包含二维平面上的点, 返回这些点构成的直线数量, max points on a line, leetcode

考虑, 水平线, 垂直线, 重合点

arraylist to array

```
Integer[] bar = foo.toArray(new Integer[foo.size()]);
```

(提醒大家一下是可以计算器的。可是 LZ 没带啊。监考的人问 LZ 延期吗，我实在不再拖了，就当时说 `take without calculator` 了。)

b. 贪吃蛇问题。就是一个 2D array. 你只能向右 和下走。并且当这个数和右边或下边的数差值为+1, -1 的时候你才能走。然后求出最长的线路。并输出条线路。

机器人, dp. `Return 1 + max(x+1,y),(x,y+1)`

- 2、一个类似围棋的题目，你和对手下围棋，你掉了一个子，对手让你放任意地方，找出围死（围棋术语：“吃”）对手最多子的 SOPT。我的理解是这样~~~这题其实我看了好久题目，总觉着题意不清~然后我就按照我的理解写了个 `dfs`~（思路类似与 `lc` 的 `word search` 但是要加个 `helper function` 去 `check` 是否被围死）。

(3) 这个也是个 `careercup` 上的原题：

```
/*
Four: Print steps of doing swap
You can swap only two consecutive elements. You have to show all steps to
convert
a string into another string (both strings will be anagrams of each other).
E.g. GUM to MUG
GUM
GMU
MGU
MUG
*/
```

```
void main()
```

```
{
```

```
    char a[] = "ACADBB123";
```

```
    char b[] = "DC1BA32BA";
```

```
    char temp;
```

```
    int n = sizeof(b);
```

```
    int j, i = 0;
```

```
    while(n > i) {
```

```
        j = i;
```

```
        printf("\n%s",a);
```

```
        while(a[j] != b[i]) {
```

```
            j++;
```

```

    }
    while(j > i) {
        temp = a[j];
        a[j] = a[j-1];
        a[j-1] = temp;
        printf("\n%s",a);
        j--;
    }
    i++;
}
}

```

要记录 steps, 可以将 MUG 转为 321, 利用 bubble sort

bfs, word ladder leetcode, queue, 找最短距离

用 hashmap 记录已经出现过的, 避免重复

2: 给一个 string 类型的 RGB 值 (rr, gg, bb) 是 16 进制的, 比较 rr 大还是 gg 大还是 bb 大, 或者两者相等大于第三者, 或者三者都相等, 就是做个判断。

Char 直接比较, 字母的 ascii 值本来就比数字大

stepping numbers

The stepping number:

A number is called as a stepping number if the adjacent digits are having a difference of 1. For eg. 8,343,545 are stepping numbers. While 890, 098 are not. The difference between a '9' and '0' should not be considered as 1.

Given start number(s) and an end number(e) your function should list out all the stepping numbers in the range including both the numbers s & e.

dfs, 构造 stepping numbers。构造不同始发数字, 不同长度的 numbers
其实也可以 dfs, 每步都记录结果

```
static void Dfs(long s, long e, long length, long num)
{
    if (length-1==0)
    {
        if (s <= num && num <=e)
            Console.WriteLine(num);
        return;
    }
    var lastDigit=num%10;
    if (lastDigit==0)
    {
        Dfs(s, e,length-1, num*10+1);
    }
    else if (lastDigit==9)
    {
        Dfs(s, e, length - 1, num*10 + 8);
    }
    else
    {
        Dfs(s, e,length-1, num*10+lastDigit-1);
        Dfs(s, e,length-1, num*10+lastDigit+1);
    }
}
```

```

static void Main(string[] args)
{
    long s = 1;
    long e = 10000000000000000;
    var sLength = Integer.toString(s).length();
    var eLength = Integer.toString(e).length();
    for (long i = sLength; i <= eLength; ++i)
    {
//no leading zero
        for (long j = 1; j < 10; ++j)
        {
            Dfs(s, e, i, j);
        }
    }
}

```

```

private static int getGCD(int a, int b)
{
    if (a == 0)
        return b;

    while (a != b)
    {
        if (a > b)
            a = a - b;
        else if (b > a)
            b = b - a;
    }
}

```

```

else
return a;
}
return a;
}

```

In 1-9 keypad one key is not working. If some one enters a password then not working key will not be entered. You have given expected password and entered password. Check that entered password is valid or not

Ex: entered 164, expected 18684 (you need to take care as when u enter 18684 and 164 only both will be taken as 164 input)

注意 18684888, the key is 8.

<http://www.careercup.com/question?id=4893522998067200>

You are given a grid of numbers. A snake sequence is made up of adjacent numbers such that for each number, the number on the right or the number below it is +1 or -1 its value. For example,

```

1 3 2 6 8
-9 7 1 -1 2
1 5 0 1 9

```

In this grid, (3, 2, 1, 0, 1) is a snake sequence.

Given a grid, find the longest snake sequences and their lengths (so there can be multiple snake sequences with the maximum length).

ij 0, 0-n, m

dp[i][j] = 1; // snake can always be of length 1

```

if (abs(grid[i][j - 1] - grid[i][j]) == 1){ dp[i][j] = max(dp[i][j], dp[i][j - 1] + 1); }
if (abs(grid[i - 1][j] - grid[i][j]) == 1){ dp[i][j] = max(dp[i][j], dp[i - 1][j] + 1); }

```

All the shoes are Reebok with the exception of two shoes , All the shoes are Nike with the exception on two shoes. All the shoes are Fubu with the exception of two shoes. How many shoes are there ?

3: reebok, nike, fubu

2: none of reebok, nike, fubu

Find the seed of a number.

Eg : $1716 = 143 * 1 * 4 * 3 = 1716$ so 143 is the seed of 1716. find all possible seed for a given number.

```
private static void seed(int i) {
    ArrayList<Integer> factor=new ArrayList<Integer>();
    for(int j=2;j<=i/2;j++){
        if(i%j==0)
            factor.add(j);
    }
    int temp=1;
    for(int number:factor){
        int temp2=number;
        temp=number;
        while(number>;0){
            temp*=number%10;
            number=number/10;
        }
        if(temp==i)
            System.out.println(temp2);
    }
}
```

