

使用说明：

1. 本代码均为G++提交。要是以纯c或者c++，输入输出的头文件一定要加全
2. 做题是小，学知识是主要的，学习代码风格，（因为题主代码风格还说的过去，虽然有些空格还是会忘打，但是整体还行）
3. 其次是模板，像我单独成函数的都是模板，要求至少会背吧。然后再灵活应用
4. 本题解只是题主的理解，可能不能估计到细节的问题讲解，跟多的需要同学们取百度学习看帖子，这是成为大佬的必经之路
5. 不懂的问题可以询问先自行求解，然后百度，最后才是询问学长作答。
6. 代码中会留有拓展问题？希望同学们试验完成，认真作答
7. 为了方便同学们学习理解我们大部分题目用了c语言去解答，（大部分小伙伴没有学过c++，建议寒假自学，基础的还是特别简单的）
8. 关于链接：本题解的链接不是友情链接，是题主有写话不想写了，或者证明的不够完善取寻找的链接，是必须要点开看的
9. 当然做题方法不知一种，我也只是选用了其中一种便于理解的进行了作答
10. 出来背锅，题目超纲（H，J），没有模板实在难以完成。我的锅。。。。承认了错误我们还是好朋友对吧，虽然我没见过你们。[捂脸]
11. 题解为一人完成，没有经过校验，有重大问题我会更正题解的。错别字见谅啦。

A -HUD 1431

题目大意

xiaoou33对既是素数又是回文的数特别感兴趣。比如说151既是素数又是个回文。现在xiaoou333想要你帮助他找出某个范围内的素数回文数，请你写个程序找出 a 跟b 之间满足条件的数。(5 <= a < b <= 100,000,000);

所用知识

回文数判断，素数判断,回文素数的特性

题目讲解

<https://zh.wikipedia.org/wiki/%E5%9B%9E%E6%96%87%E7%B4%A0%E6%95%B0>

要点：偶数位没有回文素数，任何偶数位的回文数都可以被11整除

代码实现

1. 暴力打表

```
/*
*****
> File Name: A2.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月17日 星期一 11时07分02秒
*****/

#include<iostream>
#include <stdio.h>
#define maxn 10000
using namespace std;
```

```

int arr[maxn] = {0}, times = 0;
//模板
int is_prime(int x) {
    if (x < 2) return 0;
    for (int i = 2; i * i <= x; i++) {
        if (x % i == 0) return 0;
    }
    return 1;
}
//模板
int is_pali(int x) {
    int m = x, n = 0;
    while (m) {
        n = n * 10 + m % 10;
        m /= 10;
    }
    return n == x;
}
void sum_prime() {
    for (int i = 2; i <= 100000000; i++) {
        if(is_pali(i) && is_prime(i)) arr[times++] = i;
        //拓展问题？这两个顺序有区别吗？请试验完成。
    }
    return ;
}
int main() {
    int a, b;
    sum_prime();
    while (scanf("%d%d", &a, &b) != EOF) {
        for (int i = 0; arr[i] <= b && i < times; i++) {
            if (arr[i] < a) continue;
            printf("%d\n", arr[i]);
        }
        printf("\n");
    }
    return 0;
}

```

B -HDU 1002

题目大意

计算A+b的和

所用知识

无

题目讲解

说来惭愧这道题几分钟撸完之后，提交WA了，没有除前导0，除完之后提交又WA了

说个数错误，修改了几次才知道是两个案例之间有空行，最后一个没有，我从未见过如此厚颜无耻的题，，，看了是老了，不熟悉这些题的套路了。

代码实现

```
/*
*****
> File Name: B.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月18日 星期二 11时44分45秒
*****
*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define maxn 1005

int main(){
    int n;
    char a[maxn], b[maxn], *p = a, *q = b;
    int ans[maxn] = {0}, len1 = 0, len2 = 0;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++){
        scanf("%s%s", a, b);
        p = a, q = b;
        memset(ans, 0, sizeof(ans));
        while (p[0] == '0') p += 1;
        while (q[0] == '0') q += 1;
        len1 = strlen(p); len2 = strlen(q);
        for (int i = len1 - 1, j = 1; i >= 0; i--, j++) ans[j] = p[i] - '0';
        for (int i = len2 - 1, j = 1; i >= 0; i--, j++) ans[j] += (q[i] - '0');
        ans[0] = len1 > len2 ? len1 : len2;
        for (int i = 1; i <= ans[0]; i++) {
            if(ans[i] < 10) continue;
            ans[i + 1] += ans[i] / 10;
            ans[i] %= 10;
            if(i == ans[0]) ans[0]++;
        }
        printf("Case %d:\n%s + %s = ", i, a, b);
        for (int i = ans[0]; i >= 1; i--) {
            printf("%d", ans[i]);
        }
        if (ans[0] == 0) printf("0");
        printf("\n");
        if(i != n) printf("\n");//就是这个坑死我了，同学们不要粗心啊
    }
    return 0;
}
```

C -CodeForces 998a

题目大意

找到一个由 k 个成员组成的团队，每个成员的分数不能一样

没有就打印没有，有就打印 y e s，和任意一组成员

所用知识

无

题目讲解

这里选用了标记数组（也可称为哈希）进行实现，当然方法还有其他的，没有过难的算法应该挺好懂得

代码实现

1. 哈希实现

```

/*****
> File Name: C.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月17日 星期一 16时15分19秒
*****/

#include <iostream>
#include <cstdio>
using namespace std;

int main(){
    int n, k, temp;
    int a[101] = {0};
    scanf("%d%d", &n, &k);
    for(int i = 1; i <= n; i++) {
        scanf("%d", &temp);
        if (a[temp] != 0) continue;
        a[temp] = i;
        a[0]++;
    }
    if(a[0] >= k) {
        printf("YES\n");
        for (int i = 1; i < 101 && k > 0; i++){
            if (a[i] != 0) {
                printf(k != 1 ? "%d " : "%d\n", a[i]);
                k--;
            }
        }
    } else {
        printf("NO\n");
    }
}
```

2. 思路提示也定义一个结构体存储存储下标和输入的值，对值进行排序，统计一下不同元素个数，大于k就进行筛选输出，小于就不输出，这里不进行代码演示了

D -HDU 2034

题目大意

求 $A - B$ ，也就是两个集合的差集

所用知识

这道题做法很多，可以用二分，排序，或者 `set`，本次用的是排序方法

题目讲解

本题运用了结构体加快速排序模板，对a类型标记类型为0，对1类型标记为1，按照数组大小排序，排序后，如果一个a中的一个值，左边或者右边没有相同的b中的值，则他一定存在与差集中

代码实现

```
/*  
> File Name: D.cpp  
> Author: ldc  
> Mail: litesla  
> Created Time: 2018年12月17日 星期一 18时16分19秒  
*/  
  
#include<iostream>  
using namespace std;  
#define maxn 105  
  
typedef struct Node{  
    int val,type;  
}Node;  
//快速排序模板  
void quick_sort(Node *a, int l, int r) {  
    if (l >= r) return ;  
    int x = l, y = r;  
    Node z = a[l];  
    while (x < y) {  
        while (x < y && a[y].val >= z.val) a[y--];  
        if (x < y) a[x++] = a[y];  
        while (x < y && a[x].val <= z.val) a[x++];  
        if (x < y) a[y--] = a[x];  
    }  
    a[x] = z;  
    quick_sort(a, l, x - 1);  
    quick_sort(a, x + 1, r);  
    return ;  
}  
  
int main(){  
    int n, m,ty;  
    Node a[maxn * 2];
```

```

while (scanf("%d%d", &n, &m) != EOF){
    if (n == 0 && m == 0) break;
    int times = 0;
    for (int i = 0; i < n + m; i++) {
        if(i < n) ty = 0;
        else ty = 1;
        scanf("%d", &a[i].val);
        a[i].type = ty;
    }
    quick_sort(a, 0, n + m - 1);

    for (int i = 0; i < n + m; i++) {
        if (a[i].type) continue;//b集合
        if (i - 1 > 0 && a[i].val == a[i - 1].val) continue;
        if (i + 1 < n + m && a[i].val == a[i + 1].val) continue;
        printf("%d ", a[i].val);
        times++;
    }
    if(times == 0) printf("NULL\n");
    else printf("\n");
}
return 0;
}

```

E -HDU 2035

题目大意

求 A^B 的最后三位数表示的整数。 说明： A^B 的含义是“A的B次方”

所用知识

无，知识拓展：快速幂

题目讲解

暴力防范作者道题完全能过，但是大多数这样的题是需要用快速幂来解决的。

快速幂讲解：

对用a的b次方来说，直接来说可以看成b个a相乘，例如：如果我们把 $b = 73$ ，转化成二进制数1001001，a的b次方就可以看成是 $(a^1) * (a^8) * (a^{64})$ ，代码用来按位与符号，同学们模拟一下运行的过程还是挺好理解的

百度链接：

<https://baike.baidu.com/item/%E5%BF%AB%E9%80%9F%E5%B9%82/5500243?fr=aladdin>

代码实现

1. 暴力实现

```

/*****
> File Name: E.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月16日 星期日 20时32分41秒
*****/

#include<iostream>
using namespace std;

int main() {
    int a, b, sum = 1;
    while(scanf("%d%d", &a, &b) != EOF) {
        if (a == 0 && b == 0) break;
        sum = 1;
        for (int i = 0; i < b; i++) {
            sum = (sum * a) % 1000;
        }
        printf("%d\n", sum);
    }
    return 0;
}

```

2. 快速幂实现

```

/*****
> File Name: E2.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月16日 星期日 20时38分31秒
*****/

#include<iostream>
using namespace std;
//快速幂模板
int power(int a, int b, int mod) {
    int ans = 1;
    while (b) {
        if (b & 1) ans = (ans * a) % mod;
        a = (a * a) % mod;
        b >>= 1;
    }
    return ans;
}

int main() {
    int a, b;
    while(scanf("%d%d", &a, &b) != EOF) {
        if (a == 0 && b == 0) break;
        printf("%d\n", power(a, b, 1000));
    }
}

```

```
}
```

F -HDU 2017

题目大意

对于给定的一个字符串，统计其中数字字符出现的次数。

所用知识

基本知识

题目讲解

无

代码实现

```
/*
*****
> File Name: F.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月16日 星期日 19时19分32秒
*****
*/

#include<iostream>
using namespace std;

int main(){
    int n = 0, times = 0;
    char ch[1000000];
    cin >> n;
    while (n--) {
        times = 0;
        scanf("%s", ch);
        for (int i = 0; ch[i]; i++) {
            if (ch[i] >= '0' && ch[i] <= '9') times++;
        }
        printf("%d\n", times);
    }
    return 0;
}
```

G -HDU 2020 (绝对值排序)

题目大意

输入n($n \leq 100$)个整数，按照绝对值从大到小排序后输出。题目保证对于每一个测试实例，所有的数的绝对值都不相等。

所用知识

绝对值函数，排序函数，拓展知识：快速排序

题目讲解

排序输出便可

快速排序讲解

快速排序的思想是分治，及分而治之，先取一个值，遍历一遍把数组中小于这个值的数据都放在他的左边，大于这个值的都放在他的右边，然后对左边右边递归执行上述操作，直到问题规模缩小到1，（工程中一般缩小到某个特定的值就不向下递归了）

拓展问题：快速排序的时间复杂度分析

百度百科链接：

<https://baike.baidu.com/item/%E5%BF%AB%E9%80%9F%E6%8E%92%E5%BA%8F%E7%AE%97%E6%B3%95/369842?fromtitle=%E5%BF%AB%E9%80%9F%E6%8E%92%E5%BA%8F&fromid=2084344&fr=aladin>

代码实现

```
/*
*****
> File Name: G.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月16日 星期日 19时44分11秒
*****
*/

#include<iostream>
using namespace std;
//模板算法, abs属于本题应用
void quick_sort(int *arr, int l, int r) {
    if (l >= r) return ;
    int x = l, y = r, z = arr[l];
    while (x < y) {
        while (x < y && abs(arr[y]) < abs(z)) y--;
        if (x < y) arr[x++] = arr[y];
        while (x < y && abs(arr[x]) > abs(z)) x++;
        if (x < y) arr[y--] = arr[x];
    }
    arr[x] = z;
    quick_sort(arr, 0, x - 1);
    quick_sort(arr, x + 1, r);
}

int main(){
    int n;
    int arr[100];
    while (scanf("%d", &n) != EOF) {
```

```

        if (n == 0) break;
        for (int i = 0; i < n; i++) {
            scanf("%d", &arr[i]);
        }
        quick_sort(arr, 0, n - 1);
        for (int i = 0; i < n; i++) {
            printf(i == n - 1 ? "%d\n" : "%d ", arr[i]);
        }
    }
    return 1;
}

```

H - HDU 5901 (大素数模板, 线性筛模板) 超纲

题目大意

算出 n ($1 \leq n \leq 10^{11}$) 次方的素数

所用知识

哎, 这道题主撸不出来, 本意是讲解线性筛的, 但是没注意到范围, 附加一个线性筛模板以示安慰, 竞赛中如果遇到这个题, 带好板子, , , ,

题目讲解

参考网址: https://en.wikipedia.org/wiki/Prime-counting_function 维基百科的解析

题解网址: <https://www.cnblogs.com/Ritchie/p/5886186.html> 只有代码没有解析

线性筛讲解

拓展学习: 其实介于暴力和线性筛算法时间复杂度之间的还有一个素数筛, 自行百度吧

详细讲解的百度文库: <https://wenku.baidu.com/view/4881881daaea998fcc220e99.html>

不要问题为啥不自己讲, 因为字太多了, 打的少的话难理解, , ,

线性筛可以解决的问题所有与因子有关的题, 下面是一个总结的非常好的帖子, 好好珍藏吧, 背下来更好,

<https://blog.csdn.net/Ruger008/article/details/80245687>

代码实现

```

//Meisell-Lehmer
#include<cstdio>
#include<cmath>
using namespace std;
#define LL long long
const int N = 5e6 + 2;
bool np[N];
int prime[N], pi[N];
int getprime()
{

```

```

int cnt = 0;
np[0] = np[1] = true;
pi[0] = pi[1] = 0;
for(int i = 2; i < N; ++i)
{
    if(!np[i]) prime[++cnt] = i;
    pi[i] = cnt;
    for(int j = 1; j <= cnt && i * prime[j] < N; ++j)
    {
        np[i * prime[j]] = true;
        if(i % prime[j] == 0) break;
    }
}
return cnt;
}

const int M = 7;
const int PM = 2 * 3 * 5 * 7 * 11 * 13 * 17;
int phi[PM + 1][M + 1], sz[M + 1];
void init()
{
    getprime();
    sz[0] = 1;
    for(int i = 0; i <= PM; ++i) phi[i][0] = i;
    for(int i = 1; i <= M; ++i)
    {
        sz[i] = prime[i] * sz[i - 1];
        for(int j = 1; j <= PM; ++j) phi[j][i] = phi[j][i - 1] - phi[j / prime[i]][i -
1];
    }
}

int sqrt2(LL x)
{
    LL r = (LL)sqrt(x - 0.1);
    while(r * r <= x) ++r;
    return int(r - 1);
}

int sqrt3(LL x)
{
    LL r = (LL)cbrt(x - 0.1);
    while(r * r * r <= x) ++r;
    return int(r - 1);
}

LL getphi(LL x, int s)
{
    if(s == 0) return x;
    if(s <= M) return phi[x % sz[s]][s] + (x / sz[s]) * phi[sz[s]][s];
    if(x <= prime[s]*prime[s]) return pi[x] - s + 1;
    if(x <= prime[s]*prime[s]*prime[s] && x < N)
    {
        int s2x = pi[sqrt2(x)];
        LL ans = pi[x] - (s2x + s - 2) * (s2x - s + 1) / 2;
        for(int i = s + 1; i <= s2x; ++i) ans += pi[x / prime[i]];
        return ans;
    }
}

```

```

    }
    return getphi(x, s - 1) - getphi(x / prime[s], s - 1);
}
LL getpi(LL x)
{
    if(x < N) return pi[x];
    LL ans = getphi(x, pi[sqrt3(x)]) + pi[sqrt3(x)] - 1;
    for(int i = pi[sqrt3(x)] + 1, ed = pi[sqrt2(x)]; i <= ed; ++i) ans -= getpi(x /
prime[i]) - i + 1;
    return ans;
}
LL lehmer_pi(LL x)
{
    if(x < N) return pi[x];
    int a = (int)lehmer_pi(sqrt2(sqrt2(x)));
    int b = (int)lehmer_pi(sqrt2(x));
    int c = (int)lehmer_pi(sqrt3(x));
    LL sum = getphi(x, a) + (LL)(b + a - 2) * (b - a + 1) / 2;
    for (int i = a + 1; i <= b; i++)
    {
        LL w = x / prime[i];
        sum -= lehmer_pi(w);
        if (i > c) continue;
        LL lim = lehmer_pi(sqrt2(w));
        for (int j = i; j <= lim; j++) sum -= lehmer_pi(w / prime[j]) - (j - 1);
    }
    return sum;
}
int main()
{
    init();
    LL n;
    while(~scanf("%lld",&n))
    {
        printf("%lld\n", lehmer_pi(n));
    }
    return 0;
}

```

线性筛模板

```

/*****
> File Name: is_prime.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月18日 星期二 09时48分53秒
*****/

#include<iostream>
using namespace std;
#define maxn 1000000

```

```

int prime[maxn] = {0};

int is_prime() {
    for (int i = 2; i < maxn; i++) {
        if (!prime[i]) {
            prime[++prime[0]] = i;
        }
        for (int j = 1; j <= prime[0]; j++) {
            if (prime[j] * i > maxn) break;
            prime[prime[j] * i] = 1;
            if (i % prime[j] == 0) break;
        }
    }
    return 0;
}

int main(){
    is_prime();
    printf("%d\n", prime[10001]);
    return 0;
}

```

I -HDU 1420 (快速幂)

题目大意

计算 $A^B \bmod C$ ，注意越界问题 需要开longlong型

所用知识

这道题我看了不用快速幂也没有超时，但是我们要学习的是快速幂，所以就用快速幂讲解，附加某个同学通过的代码 (做了格式上的一些修改)，题主偷懒不想撸了

题目讲解

上面讲过快速幂了，不懂的请遵循使用说明第5条。

代码实现

1. 快速幂算法 提交用时15ms

```

/*****
> File Name: I.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月18日 星期二 10时00分10秒
*****/

#include<iostream>
using namespace std;
//模板

```

```

long long power(long long a, long long b, long long mod) {
    long long ans = 1;
    while (b) {
        if (b & 1) ans = ans * a % mod;
        a = a * a % mod;
        b >>= 1;
    }
    return ans;
}

int main(){
    long long n;
    long long a, b, c;
    scanf("%lld", &n);
    while (n--) {
        scanf("%lld%lld%lld", &a, &b, &c);
        printf("%lld\n", power(a, b, c));
    }
    return 0;
}

```

2. 某个同学的暴力算法 提交用时140ms

```

#include<iostream>
using namespace std;
int main() {
    int n;
    cin>> n;
    while(n-->0) {
        int a, b, c;
        cin>> a >> b >> c;
        long long int sum = 1;
        for(int i = 1; i <= b; i++)
        {
            sum = sum * a % c;
            if(sum == 0)
                break;
        }
        cout<< sum <<endl;
    }
    return 0;
}

```

根据提交用时我们就可以看出速度的提升了

J -POJ 2756 (高精度)

题目大意

秋天是个天才可以做加减法，但是数有最大值，没有最小值，

所用知识

高精度加减法

题目讲解

直接模拟吧，，

附加：小神童太神了，，，我2岁时可能才会说几个字，路都不一定会走，别说计算了，数我都不认识。

再次附加：这道题用java会特别简答，但是这道题用Java会特别简单，，，，但是不提倡。先苦后甜嘛

代码实现

//这道题，我承认我百度的板子，，我刚开始打了打，但是实在是不优美，又臭又长，而且我也没有专业学习过高精度的算法，估计要找半天bug，时间有限。所以还是去寻找大佬的帖子了，这题我出的，这锅我背了，，这句话藏这么深估计看到的人很少吧，hahaha

//大佬的链接：<https://blog.csdn.net/kenden23/article/details/40449333>

```
#include <stdio.h>
#include <string>
#include <algorithm>
using std::string;

const int MAX_B = 5120;
char buf[MAX_B];
int id = 0, len = 0;

inline char getFromBuf()
{
    if (id >= len)
    {
        len = fread(buf, 1, MAX_B, stdin);
        id = 0;
    }
    return buf[id++];
}

void getIntFromBuf(string &n)
{
    char a = getFromBuf();
    while ((a == ' ' || a == '\n') && len) a = getFromBuf();

    bool sign = true;
    if (a == '-' || a == '+')
    {
        if (a == '-') sign = false;
        a = getFromBuf();
    }
    n.clear();
    while ((a != ' ' && a != '\n') && len) //老是写&&, 错成||
    {
        n.push_back(a);
        a = getFromBuf();
    }
}
```

```

    }
    if (sign) n.push_back('+');
    else n.push_back('-');
}

string operator+(string &a, string &b)
{
    string c;
    int N1 = (int)a.size(), N2 = (int)b.size();
    int carry = 0;
    for (int i = N1-1, j = N2-1; i>=0 || j>=0 || carry; i--, j--)
    {
        int an = i>=0? a[i]-'0' : 0;
        int bn = j>=0? b[j]-'0' : 0;
        int sum = an + bn + carry;
        carry = sum / 10;
        c.push_back(sum % 10 + '0');
    }
    reverse(c.begin(), c.end());
    return c;
}

string operator-(string &a, string &b)
{
    string c;
    int N1 = (int)a.size(), N2 = (int)b.size();
    int carry = 0;
    for (int i = N1-1, j = N2-1; i>=0 || j>=0 || carry; i--, j--)
    {
        int an = i>=0? a[i]-'0' : 0;
        int bn = j>=0? b[j]-'0' : 0;
        int sum = an - bn + carry;
        if (sum < 0)
        {
            carry = -1;
            sum += 10;
        }
        else carry = 0;
        c.push_back(sum % 10 + '0');
    }
    reverse(c.begin(), c.end());
    return c;
}

int cmpAbsStr(string &a, string &b)
{
    if (a.size() < b.size()) return -1;
    else if (a.size() > b.size()) return 1;
    if (a == b) return 0;

    for (int i = 0; i < (int)a.size(); i++)
    {
        if (a[i] < b[i]) return -1;
    }
}

```



```

        else if (a[i] > b[i]) return 1;
    }
    return 0;
}

int main()
{
    int T;
    scanf("%d", &T);
    getchar();
    string n1, n2;
    while (T--)
    {
        getIntFromBuf(n1);
        getIntFromBuf(n2);
        if (n1[n1.size()-1] == '+' && n2[n2.size()-1] == '+'
            || n1[n1.size()-1] == '-' && n2[n2.size()-1] == '-')
        {
            if (n1[n1.size()-1] == '-' && n2[n2.size()-1] == '-') putchar('-');
            //n1.pop_back();n2.pop_back();
            n1.erase(n1.size()-1); n2.erase(n2.size()-1);
            string c = n1 + n2;
            puts(c.c_str());
        }
        else
        {
            if (n1[n1.size()-1] == '-' && n2[n2.size()-1] == '+') n1.swap(n2);
            //n1.pop_back(); n2.pop_back();
            n1.erase(n1.size()-1), n2.erase(n2.size()-1);

            int sign = cmpAbsStr(n1, n2);
            if (sign == 0) puts("0");
            else if (sign == 1)
            {
                string c = n1 - n2;
                puts(c.c_str());
            }
            else
            {
                string c = n2 - n1;
                putchar('-');
                puts(c.c_str());
            }
        }
    }
    return 0;
}

```

K - HDU 1022 (栈的简单应用)

题目大意

列车按O1的顺序进入铁路，你的任务是确定列车是否可以在订单O2中出来。

所用知识

栈的模型（先进后出）

题目讲解

模拟题，当栈为空时，或者当第一个序列的元素不等于第二个元素的下标的时候进栈，当等于的时候出栈，出栈，当存储的元素大于第一个序列的元素数量，还在存储的时候，说明错误，直接返回，栈的函数可以直接使用stl，但是有些同学可能不了解，所以直接用了数组和一个head指针代替了

代码实现

```

/*****
> File Name: K.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月18日 星期二 16时55分46秒
*****/

#include<iostream>
#include <string.h>
using namespace std;
#define maxn 15

int main(){
    int a[maxn * 2], c = 0;
    int stack[maxn], head = -1, i, j;
    char str1[maxn], str2[maxn];
    while (scanf("%d%s%s",&c, str1, str2) != EOF) {
        a[0] = 0, head = -1;
        i = 0, j = 0;
        while (j < c) {
            if (head < 0 || stack[head] != str2[j]) {
                if(i == c) break;
                stack[++head] = str1[i];//进栈
                a[++a[0]] = 1;//标记
                i++;//数组后移
            } else {
                head--;//出栈
                a[++a[0]] = 2;//标记
                j++;//数组后移
            }
        }
        if (head == -1) printf("Yes.\n");
        else printf("No.\n");
        for (int i = 1; head == -1 && i <= a[0]; i++) {
            if(a[i] == 1) printf("in\n");
            else printf("out\n");
        }
        printf("FINISH\n");
    }
    return 0;
}
```

L -POJ 3032 (队列的简单应用)

题目大意

一个魔术师第一次拿一章牌放到下面，然后拿出黑桃一，第二次拿出两张放到下面拿出黑桃2，以此类推，到n，问怎样排序才可以达到这样的效果

所用知识

队列的简单应用（先进先出），反向模拟

题目讲解

这道题直接模拟的话，会收到后面的影响，但是如果反向模拟的话就不会有这中影响，我们的队头对应的是牌底，队尾对应的是牌顶，注意翻牌操作也是反向的，在大脑中想想一个倒放的画面就会很好理解。

注意：为了方便我直接开了一个大队列，就不用循环队列了，，，

代码实现

```
/*
*****
> File Name: L.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月18日 星期二 21时05分09秒
*****
*/

#include<iostream>
#include <cstdio>
using namespace std;

#define maxn 10000

int main(){
    int queue[maxn],head = 0,tail = -1;//head队头 tail队尾
    int n = 0,temp;
    scanf("%d", &n);
    while (n--) {
        scanf("%d", &temp);
        queue[++tail] = temp;// 进队
        for (int i = temp - 1; i >= 1; i--) {
            queue[++tail] = i;//进队
            for (int j = 1; j <= i; j++) {
                int k = queue[head++];//出队
                queue[++tail] = k;//进队
            }
        }
    }
    while (head <= tail) {
        printf("%d ", queue[tail--]); //逆序输出
    }
}
```

```
        printf("\n");
    }
    return 0;
}
```

M -HDU 2010

题目大意

求100到999之间的水仙花数

所用知识

无

题目讲解

注意输出格式，相对简单，也没有固定的套路，需要看代码能力了

代码实现

```
/*
*****
> File Name: M.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月17日 星期一 22时06分51秒
*****
*/

#include<iostream>
using namespace std;
#define maxn 1000

int arr[maxn] = {0};
int f(int x) {
    int m = 0, n = 0;
    while (x) {
        n = x % 10;
        m += n * n * n;
        x /= 10;
    }
    return m;
}

void init(){
    for (int i = 100; i < maxn; i++) {
        if(f(i) == i) arr[++arr[0]] = i;
    }
    return ;
}

int main(){
    int n ,m, times = 0;
```

```

init();
while (scanf("%d%d", &n, &m) != EOF) {
    times = 0;
    for (int i = 1; arr[i] <= m && i <= arr[0]; i++) {
        if (arr[i] < n) continue;
        printf("%d",arr[i]);
        if(i + 1 <= arr[0] && arr[i + 1] <= m) {
            printf(" ");
        }
        times++;
    }
    if (times)printf("\n");
    else printf("no\n");
}
return 0;
}

```

N -HDU 2072

题目大意

有多组数据，每组一行，每组就是一篇小文章。每篇小文章都是由小写字母和空格组成，没有标点符号，遇到#时表示输入结束。

所用知识

题目用c语言实现起来过于复杂，所以题主（这道题不是我出的不要恨我）用c++来实现解答了

题目讲解

while会读取一个字符，str用来暂时存储每个单词，当一个单词读取完毕时，存储到 `set<string>` 中，set的内部实现是红黑树(红黑树的内容超纲过于严重，现在这个阶段你们只需要会使用就可以了)，简单理解他的作用是不会存储相同的单词，且查找时间负责度低。

红黑树: https://blog.csdn.net/Yu_L2/article/details/79306000

代码实现

```

/*****
> File Name: N.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月17日 星期一 19时20分09秒
*****/

#include <iostream>
#include <string>
#include <set>
using namespace std;

```

```

int main(){
    char w;
    string str;
    set<string> word;
    while ((w = getchar()) != '#') {
        if(w >= 'a' && w <= 'z') str.push_back(w);
        if(w == ' ' && str.size() > 0) {
            word.insert(str);
            str.clear();
        }
        if(w == '\n') {
            if(str.size() > 0) {
                word.insert(str);
                str.clear();
            }
            cout << word.size() << endl;
            word.clear();
        }
    }
    return 0;
}

```

O -HDU 2015

题目大意

有一个长度为 n ($n \leq 100$) 的数列，该数列定义为从2开始的递增有序偶数，现在要求你按照顺序每 m 个数求出一个平均值，如果最后不足 m 个，则以实际数量求平均值。编程输出该平均值序列。

所用知识

无， 注意输出格式，行末位没有空格

题目讲解

水题，不讲了，但是你们可以看我的代码。

代码实现

```

/*****
> File Name: O.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月17日 星期一 18时57分04秒
*****/

#include<iostream>
using namespace std;

int main(){
    int n, m, in = 2, times = 0, sum = 0;

```

```

while (scanf("%d%d", &n, &m) != EOF) {
    times = 0, sum = 0;
    for (int i = 2; n; i += 2, n--) {
        sum += i;
        times++;
        if(times == m) {
            printf("%d", sum / times);
            sum = 0;
            times = 0;
            n != 1 && printf(" ");
        }

    }
    if(times != 0) printf("%d\n", sum / times);
    else printf("\n");
}
return 0;
}

```

P -HDU 2000

题目大意

输入三个字符后，按各字符的ASCII码从小到大的顺序输出这三个字符

所用知识

ASCII 编码常识

题目讲解

水题，不讲了 但是你们可以看我的代码。

代码实现

```

/*****
> File Name: P.cpp
> Author: ldc
> Mail: litesla
> Created Time: 2018年12月16日 星期日 19时05分39秒
*****/

#include<iostream>
//模板
#define swap(a, b) ({\
    __typeof(a) _temp = a;\
    a = b; b = _temp;\
})
using namespace std;

int main() {

```

```

char str[3];
while (cin >> str) {
    str[0] > str[1] ? swap(str[0], str[1]) : 0; //可以换成if语句
    str[1] > str[2] ? swap(str[1], str[2]) : 0;
    str[0] > str[1] ? swap(str[0], str[1]) : 0;
    printf("%c %c %c\n", str[0], str[1], str[2]);
}
return 0;
}

```

拓展问题—提示

A题代码部分有一个小试验，定义一个全局变量作为计数器，在每个函数内调用，然后交换两个函数的顺序，看计数器的统计次数就知道了

G题有一个快速排序时间复杂度分析，快速排序是缩小问题规模进行求解，冒泡是 n^2 当 n 缩小一半排序时 $(n/2)^2 + (n/2)^2 = (n^2)/2$等等等等可以继续往下细分

赠与有缘人的彩蛋：刚下学期开始会有比赛，下学期刚开始会有比赛，下学期刚开始会有比赛，暂定知识点二分，栈，队列，快速幂，线性筛，排序，可能会有原题。小伙伴们寒假好好准备啊。爆零者，嘿嘿嘿。