

写在前面：

应 zjl（群里的群文件小助手 22 电气 看我头衔）之邀，写 2020 年网安机考这几个题的题解以及机考的一些准备思路。

题目难度方面，我基本是按洛谷难度来评判的，四道题都不到普及组难度，暑假一个月冲刺或者前半学期慢慢学都没什么问题，不必焦急。考试时间 2.5h，写四个暴力绰绰有余。参加机考的前提是先过笔试，然后面试的时候也挺看重数学成绩的，不要因为太过重视机考而忽视数学英语。

——陶一刀留

同上，笔试分数往往更占主要部分，20 级机考极差很大，大部分都是两位数分数。数学跟英语都蛮重要。如何在大一上入学适应阶段和课程压力下保持转专业的复习状态并且兼顾本专业课程对不少人来说才是最大的难点。

——对角线留

题解：

T1 求 LCM(a,b)

最小公倍数

Description

给定两个正整数a和b，计算a和b的最小公倍数。

Input

一行，两个正整数a,b，用空格隔开

Output

一行，一个整数，表示a和b的最小公倍数。

Sample Input 1

4 6

Sample Output 1

12

Hint

$1 \leq a, b \leq 1e9$

*本评测系统打印long long型整数使用"lld%"

S

显然 a 和 b 的最小公倍数是 $a*b/\gcd(a,b)$ (a 和 b 的最大公因数)。
辗转相除法求 $\gcd(a,b)$ 。

Example:

```
//
```

```
int GCD( int a , int b ){
```

```
if(b==0) return a;
```

```
return GCD(b,a%b);
```

```
}
```

```
//
```

时间复杂度: $O(\log a)$

难度：入门难度（高中数学知识，不会的可以去面壁了）

T2 报数

报数

Description

n 个同学围成一个圈，按顺时针方向编号依次为 1, 2, ..., n ，从编号为 1 的同学开始顺时针报数，1 号同学首先报 1，接下来的同学报数依次加 1，碰到 7 的倍数，或者报数中含有数字 7，那么这个同学退到圈外，直到最后只剩下一个同学，要求该同学的编号。

Input

一行，一个正整数 n

Output

一行，一个正整数，表示最后剩下的同学的编号。

Sample Input 1

5

Sample Output 1

1

暴力想法：1~ n 表示 1~ n 号同学，用 vis 数组表示有没有退圈($vis[i]=1$ 代表 i 同学已经退圈, $vis[i]=0$ 代表 i 同学还未退圈)。用计数器 cnt 记录有多少人已退圈。然后按照题目描述进行模拟。退圈人数不到 $n-1$ 个时 (While($cnt!=n-1$))，按他描述的方法轮流报数，1 (i) 号同学报 1(j), ..., 直到 n 号同学报完再从 1 号同学开始循环 ($if(i==n+1) \{i=1\}$)，期间如果 i 号同学报的数 j 满足了上面的条件, 就让 i 号同学退圈($vis[i]=1$); 并记录退圈人数+1($cnt++$)。之后循环到 i 时，如果 i 已经退圈($vis[i]=1$)，我们就跳过 i 号让 $i+1$ 号继续报数，直到退圈人数(cnt)等于 $n-1$ 为止

Example:

//

int cnt=0;//计数器，记录已经有多少人退圈

int i=1,j=1;//目前第 i 个人报数,应该报 j 号

while(cnt!=n-1){//退圈人数不到 $n-1$ 个

if($i==n+1$) $\{i=1\}$;//已经报完一轮，重新从第 1 个开始

if($vis[i]==1$) $\{i++$;continue;}//第 i 个人已经退圈了，让下一个 ($i+1$ 号) 继续报

//现在 i 号同学报 j

if(j 满足上面的条件) $\{j$ 满足上面的条件, i 就该退圈了

$vis[i]=1$;//记录 i 已经出圈

$cnt++$;//出圈人数+1

}

$j++$;//下一个人报 $j+1$ 号

}//退圈人数(cnt)到 $n-1$ 个时,场上只剩 1 个人了,停止报数

for(int $i=1$; $i \leq n$; $i++$)

if($vis[i]==0$) $ans=i$;// 如果 i 没有出圈， i 就是最后留下的这个人

//

时间复杂度: $O(n)$

难度：入门难度（本质上是个约瑟夫环，裸模拟就完事，Oler 不会的自觉去面壁）

优化思路：

1.链表（用链表写理解起来会容易很多，实现也简单，显然可行）

2.尝试递推（没试过，不保证可行性）

T3 寻找最小值

寻找最小值

Description

给定一个长度为 n 的有序数列， $a_1 \leq a_2 \leq a_3 \leq \dots \leq a_n$;

请你将这个数列划分为 m 组，每组的定义为连续的几个数 $a_i, a_{i+1}, \dots, a_{i+k}$ ，每组的权值为其最大值与最小值的差 x_i ，设 m 组的权值和为 X ，

求 X 的最小值。

Input

第一行两个整数 n, m ;

第二行 n 个整数， $a_1, a_2, a_3, \dots, a_n$ 。

Output

一个整数表示 X 。

Sample Input 1

```
5 2
1 3 4 6 7
```

Sample Output 1

```
4
```

暴力想法： $a[1]$ 到 $a[n]$ 分成 m 组，求每组极差之和的最小值

a 数组单增，对于每一组，极差显然是这一组最后一个减这一组最前面一个。

例：从 $a[i]$ 处划分开： $(a[1], a[2], \dots, a[i]), (a[i+1], \dots, a[n-1], a[n])$

第一组极差为 $a[i] - a[1]$ ，第二组极差为 $a[n] - a[i+1]$ ，极差和为 $X = a[n] - a[i+1] + a[i] - a[1]$ ，即：
 $X = a[n] - a[1] + \sum(a[i] - a[i+1])$

要让 X 最小，显然要让 $m-1$ 对 $(a[i] - a[i+1])$ 最小。那么，求出所有的 $(a[i] - a[i+1])$ ，然后选最小的 $m-1$ 个就行了。用 $b[i]$ 表示 $a[i] - a[i+1]$ ，取最小的 $m-1$ 个 $b[i]$ ，加上 $a[n] - a[1]$ 即为最小 X 的值。

Example:

```
//
```

```
for(int i=1;i<n;i++) {b[i]=a[i]-a[i+1];}
```

```
sort(b+1,b+n)//b[1]到b[n-1]排序
```

```
X=a[n]-a[1];
```

```
for(int i=1;i<m;i++) X+=b[i];
```

```
//
```

时间复杂度： $O(n \log n)$ (主要是排序消耗)

难度：普及-（明显的贪心，Oler不会的…还是去面壁吧，可以比前两个少面壁一会儿）

优化思路：

1. 显然 m 小于 n ，我们可以用数据结构只记录前 m 小的 $(a[i] - a[i+1])$ ，时间复杂度 $O(n \log m)$ (显然可行)

T4 寻找最小值+

寻找最小值+

Description

给定一个长度为 n 的数列, $a_1, a_2, a_3, \dots, a_n$;

请你将这个数列划分为 m 组, 每组的定义为数列中连续的几个数 $a_i, a_{i+1}, \dots, a_{i+k}$, 每组的权值为其最大值与最小值的差 x_i , 设 m 组的权值和为 X 。

求 X 的最小值。

Input

第一行两个整数 n, m

第二行 n 个整数, $a_1, a_2, a_3, \dots, a_n$

Output

一个整数表示 X

Sample Input 1

```
5 1
1 2 3 20 5
```

Sample Output 1

```
17
```

暴力想法: a 数组不再具有特殊性质, 显然无法照搬上一道题的思路。加上如果在某处划分组显然会对后续造成影响, 所以不能贪心。于是我们选择动态规划。

设 $dp[i][j]$ 表示前 i 个划分 j 组能得到的最小权值和, 那么, 我们必定是选择前 k 个划分 $j-1$ 组, 然后把 $k+1$ 到 i 划成一组, 即:

$$dp[i][j] = \min(dp[k][j-1] + \text{极差}[k+1, i])$$

得到状态转移方程。最后求出的 $dp[n][m]$ 即为前 n 个划分 m 组时的最小权值和, 即为 X 。

再: 因为 n 非常小所以可以用二维数组 $q[i][j]$ 表示 i 到 j 的极差, 可以用 $O(n^2)$ 的时间预处理

Example:

```
//
```

预处理 q 数组//具体懒得写了, 上一题有序就帮我们节省了预处理时间

```
for(int i=1;i<=n;i++){
```

```
    for(int j=2;j<=m;j++){//思考为什么不从 1 开始(注意  $dp[k][j-1]$  的意义哦)
```

```
        for(int k=1;k<i;k++){
```

```
             $dp[i][j] = \min(dp[i][j], dp[k][j-1] + q[k+1][i]);$ 
```

```
//
```

时间复杂度: $O(n^2m)$ (不预处理 q 数组就会变成 $O(n^3m)$, 可能超时)

难度: 普及-/普及 (根据题目要求很容易想到的 dp 思路)

补充: 关于 q 数组和 dp 数组的预处理——特判: $q[i][i]=0, dp[i][i]=0$ 边界: $dp[i][1]$, 初始化:

dp 极大值初始化

优化思路:

1.st 表求区间极差 (显然可行)

2.忘情水二分(带权二分), 可优化到 $O(nm\log n)$ (不保证可行性)

准备：

按照 NOIP 普及组的方式准备就行，至少这几道题的知识点都是普及组需要的知识储备，可以参考知乎 (<https://zhuanlan.zhihu.com/p/76721044>)。Oler 上 luogu 找两道普及组的保持一下手感就行了。

0 编程基础的话：

1. 可以选择购买入门书

远古版本的《信息学奥赛一本通》，还有近几年新出的《深入浅出程序设计竞赛 - 基础篇》或者任何一本书，尽量在暑假掌握 C++ 基础语法。(卷起来了呜呜呜—对角线)

2. 注册 OJ 并进行练习

OJ (Online Judge, 在线测评系统) 是 Oler 在学习中最常用的资源与平台，大部分 OJ 均为免费并且提供了非常好的题目资源。洛谷([luogu.com.cn](https://www.luogu.com.cn))对新手挺友好的，上面的《深入浅出程序设计竞赛 - 基础篇》也是洛谷出的。(推荐书与题单结合食用—对角线)

3. 充分利用网络资源自学知识点

OI-Wiki 是一个 OI 的知识整合站点，其中的语法部分写的非常不错，建议参考学习 (<https://oi-wiki.org/>)。别的还有各种博客，想学什么知识点百度基本都能找到。(OI-wiki 偏竞赛方面，时间有限可以不看—对角线)

(*均利益无关，有些是我们自己用着比较顺手，有些是看其他人用感觉不错)

4(*). 找学长学姐，Oler 或者 C/C++ 老师

看见群里那个 20-机转网-对角线大佬了吗，学习遇到难题建议问他，数学/英语/计算机/物理啥都行。——陶一刀 (一本正经.jpg)

爬 (指楼上)。

——对角线

我不是 oier，也完全是入学了才接触编程与算法，肯定赶参赛选手有很大差距，但跟我情况一样的学弟学妹不要有太大压力，按部就班把知识学扎实就已经超过很多人了，而且不要上来就瞄准竞赛标准要求自己，不要看别人跑的太快就焦虑。算法主要需要刷题，贵在坚持与毅力。静心感受算法的魅力吧！——对角线

写在最后：

因为机考新开，20 级绝大部分都没什么准备，把 T1 的分拿满都到 Rank10 左右了，重心还是投入笔试，刷好数学。今年可能大家都有准备了，整体水平应该有一定的提高，该怎么在这些科目上分配时间，还需要各位自己把握。

祝同学们都能转到自己想进的专业。