

杭州养正学校2019年信息学奥赛团队

初一年级选拔赛试题

Statement, Code, Data By QQYZ

一、试题说明

本次考试共有8题，每题100分，满分800分，各题信息如下：

题目编号	1	2	3	4
题目中文名	能力类别	乞丐者	信息维护	螺旋方阵
题目英文名	<i>ability</i>	<i>beggar</i>	<i>defend</i>	<i>helical</i>
输入文件名	<i>ability.in</i>	<i>beggar.in</i>	<i>defend.in</i>	<i>helical.in</i>
输出文件名	<i>ability.out</i>	<i>beggar.out</i>	<i>defend.out</i>	<i>helical.out</i>
源程序文件名	<i>ability.cpp</i>	<i>beggar.cpp</i>	<i>defend.cpp</i>	<i>helical.cpp</i>
测试点数量	10		20	
单测试点分值	10		5	
评测方式	全文比较（忽略行末空格和文末回车）			
运行内存限制	256 MB			
运行时间限制	1.0 s			
题目编号	5	6	7	8
题目中文名	字符寻找	智能标记	素食主义者	拉链集合
题目英文名	<i>search</i>	<i>tab</i>	<i>vegetarian</i>	<i>zip</i>
输入文件名	<i>search.in</i>	<i>tab.in</i>	<i>vegetarian.in</i>	<i>zip.in</i>
输出文件名	<i>search.out</i>	<i>tab.out</i>	<i>vegetarian.out</i>	<i>zip.out</i>
源程序文件名	<i>search.cpp</i>	<i>tab.cpp</i>	<i>vegetarian.cpp</i>	<i>zip.cpp</i>
测试点数量	10	20		25
单测试点分值	10	5		4
评测方式	全文比较（忽略行末空格和文末回车）			
运行内存限制	256 MB			
运行时间限制	1.0 s			

二、注意事项

1. 本次比赛采用文件输入输出、Lemon软件评测。以题目*ability*为例，选手需要在源程序中*main()*函数的开头加上如下两句代码：

```
freopen("ability.in", "r", stdin);
freopen("ability.out", "w", stdout);
```

当然选手也可以采用其他的文件输入输出方式，但不保证正确性。任何由于非系统原因引起的错误都由选手负责；

2. 本次比赛统一使用 C++ 语言，且*main()*函数的类型必须为*int*，返回值必须为0，选手的源程序不得使用任何以下划线开头的系统函数；

3. 比赛前只下发本试题，比赛结束后会下发*Standard Code*，需要*Solution*的选手可向 QQYZ 提出要求。

1.能力类别 (*ability.cpp*)

【题目背景】

本*statement*中涉及到很多和BYY相关的文字，特声明如下：

BYY非常猥琐，*statement*中提到她的唯一目的是讽刺、羞辱她。

QQYZ 曾经在NOIP2018中爆过0，这导致他在之后相当长的一段时间内都不敢对外宣称自己是一个OIer。由此可见，QQYZ 堪称全养正OI实力最弱的人。但是由于一些玄学原因，他总是可以通过一些特殊的方式使得自己在模拟赛中取得一个相对正常的成绩。

这些特殊的方式，被他称作超能力，简称能力。

【题目描述】

从上面的背景可以看出，QQYZ 拥有很多不同的能力。

假设 QQYZ 拥有5种能力，且它们的特征分别如下：

- 能力1：特征值 x 是3的倍数；
- 能力2：特征值 x 不是3的倍数，且可表示为 $4k + 1 (k \in N^*)$ 的形式；
- 能力3：特征值 x 不是3的倍数，且可表示为 $5k + 2 (k \in N^*)$ 的形式；
- 能力4：特征值 x 既是4的倍数，又是7的倍数，且不是3334的约数；
- 能力5：特征值 x 的个位只能是1,3,4，且 x 是6的倍数。

现在输入给出能力的特征值 x ，试判断该能力可能是哪些类别的能力。

【输入格式】

输入文件*ability.in*包括一行，共一个正整数 x 。

【输出格式】

输出文件*ability.out*包括一行，共若干个正整数，分别表示该能力可能是哪些类别的能力，从小到大输出，每两个正整数中间有一个空格。若该能力不可能是任意一种能力，则输出"*Beautiful BYY*"（不含引号）。

【输入输出样例1】

<i>ability.in</i>	<i>ability.out</i>
17	2 3

【样例说明】

17不是3的倍数，不满足条件1； $17 = 4 \times 4 + 1 = 5 \times 3 + 2$ ，满足条件2,3；17既不是4的倍数，也不是7的倍数，不满足条件4；17的个位数为7，且不是6的倍数，不满足条件5。故该能力分别可能为能力2,3。

【输入输出样例2】

<i>ability.in</i>	<i>ability.out</i>
3334	Beautiful BYY

【数据范围与测试点信息】

对于全部数据，有 $x \leq 10000$ 且 $x \in N^*$ 。

2. 乞讨者 (*beggar.cpp*)

【题目背景】

QQYZ 的英语非常差，这导致他的综合排名也非常尴尬。

倘若他有机会说英语，他一定会说：

My English is really poor.

然而他最近发现，说这句话的人会面临一种进退维谷的境地：这是让与你交谈的人直接施舍给你一些英语，还是直接结束交谈呢？

对于一些重要场合，或许第一种解释更加合适。既然如此，QQYZ 就离奇地变成了一个乞讨者。虽然他不愿意接受这个称号，但是这是非系统原因所致，责任要自行承担（英语差的后果）。

【题目描述】

QQYZ 只在第 L 天到第 R 天之间说这句话（包括左右端点）。假设他在第 i 天说这句话，那么他所能获得的英语知识量为 i^2 。

但在某些日子，他并不能说这句话（假设当天为第 i 天）：

- 若第 i 天的编号的个位数为 6，则停止交谈；
- 若到第 $i - 1$ 天为止，已经获得的英语知识量对 998244353 取模的结果为 k 的倍数（不考虑取模结果为 0 的情况），则停止交谈；
- 若 i 是 82187891047 的非 1 约数，则停止交谈。

现在给出 L 、 R 和 k 的值，求出 QQYZ 在这段时间内可以被施舍多少英语。

【输入格式】

输入文件 *beggar.in* 包括一行，共三个正整数，分别为 L, R, k 。

【输出格式】

输出文件 *beggar.out* 包括一行，共一个正整数，表示 QQYZ 能被施舍的英语的总量。由于该总量可能很大，因此要对 998244353 取模。若 QQYZ 没有被施舍到任何英语，则输出 "Beautiful BYY"（不含引号）。

【输入输出样例1】

<i>beggar.in</i>	<i>beggar.out</i>
2 9 12	248

【样例说明】

第 2 天末总量为 4，第 3 天末总量为 13，第 4 天末总量为 29，第 5 天末总量为 54；第 6 天停止交谈；第 7 天末总量为 103，第 8 天末总量为 167，第 9 天末总量为 248。没有任何一天出现总量为 12 的倍数或 82187891047 的非 1 约数的情况。

【输入输出样例2】

<i>beggar.in</i>	<i>beggar.out</i>
2016 2017 14	Beautiful BYY

【数据范围与测试点信息】

对于全部数据，满足 $L, R \leq 10^5$ ， $k \leq 10^9$ 。

对于测试点 1 - 3，满足 $L, R \leq 100$ ， $k = 998244353$ ；

对于测试点 4 - 5，满足 $L, R \leq 10000$ ， $k = 506267$ 。

3.信息维护 (*defend.cpp*)

【题目背景】

在杭州养正学校的信奥团队中，有一种特殊的身份——信奥大班长。该职务一般由初二年级中OI实力最强、威信最高的人担任，例如*dst*。

众所周知，*dst*是前信奥大班长（2018.9 – 2019.6），经常出一些毒瘤题虐QQYZ，不像QQYZ只能出模板题送AC；*cyn*则是将要就任的信奥大班长。

很显然，QQYZ没有任何做信奥大班长的希望，但是他可以通过和现任和前任的信奥大班长搞好关系以避免被所有人虐，所以他经常和*dst*、*cyn*互相发送信息，使自己看起来没有那么弱。

【题目描述】

某一天，*dst*向QQYZ发送了一段信息，长度为*N*；*cyn*也发送了一段信息，长度为*M*。因为*dst*和*cyn*表面毫无冲突而私下矛盾重重，所以他们发送的信息也有很大一部分是针锋相对的。

QQYZ对此无能为力，因为如果让两位神犇中的任意一位看到对方发的信息中反对自己的部分，两人的信息就都要被删掉。

已知*dst*的信息由正整数组成，且其反对*cyn*的部分是一个严格单调递增的最长连续子序列；*cyn*的信息也由正整数组成，且反对*dst*的部分是一个严格单调递减的最长连续子序列。

QQYZ想要得知两人的信息中反对对方部分的长度分别是多少。

给出*N*、*M*和两人发的信息，求出两人分别反对对方的信息长度。

【输入格式】

输入文件*defend.in*包括三行，第一行为两个正整数*N*和*M*；第二行为*N*个正整数 a_i ，表示*dst*发的信息；第三行为*M*个正整数 b_i ，表示*cyn*发的信息。

【输出格式】

输出文件*defend.out*包括一行，共两个正整数，分别表示*dst*和*cyn*发的信息中反对对方部分的长度。

【输入输出样例】

<i>defend.in</i>	<i>defend.out</i>
5 7 1 3 4 2 5 2 6 4 3 5 6 4	3 3

【样例说明】

*dst*发的信息中严格单调递增的最长连续子序列为1,3,4，长度为3；*cyn*发的信息中严格单调递减的最长连续子序列为6,4,3，长度也为3。

【数据范围与测试点信息】

对于全部数据，满足 $N, M \leq 10^5$ ， $a_i, b_i \leq 10^9$ 。

对于测试点1 – 6，满足 $N, M \leq 100$ ， $a_i, b_i \leq 10^5$ ；

对于测试点7 – 14，满足 $N, M \leq 1000$ 。

4.螺旋方阵 (*helical.cpp*)

【题目背景】

QQYZ 没有任何思考能力，因此他在模拟赛的时候经常做一些*magic*的事情。例如，他很喜欢研究各种方阵，尤其喜爱螺旋方阵。

然而，螺旋方阵也有非常多种不同的类型，这非常令他头疼。不过他最近研究出了一种集各种方阵性质于一身的特殊螺旋方阵，请看下文。

【题目描述】

QQYZ 所研究出的这种特殊螺旋方阵满足如下条件：

- 方阵是 $N \cdot N$ 的矩阵， N 为偶数，但不是4的倍数；
- 方阵的上下两部分关于第 $\frac{N}{2}$ 行和 $\frac{N}{2} + 1$ 行的交界线轴对称。

现在我们以上面的部分为例，探究方阵的构造方法：

- 1.在方阵的左上角填上数字1，并将指针设在此处；
- 2.指针不断向右移动，在此过程中每格填上的数字为上一个数字加上1；
- 3.指针无法向右移动后，向下移动，填数字的方法不变；
- 4.若指针按当前方向继续走就要走出边界或者走到已填好的数字里，就按当前方向右转，填数字的方法不变（按当前方向右转的意思：原来向右则改为向下，原来向下则改为向左，原来向左则改为向上，原来向上则改为向右）；
- 5.当指针向任意方向移动都要走到已填好的数字里时，构造结束。

显然下面的部分也可以类似构造。

现在给出 N 的值，试按上面的构造方法输出这个方阵。

【输入格式】

输入文件*helical.in*包括一行，共一个正整数 N 。

【输出格式】

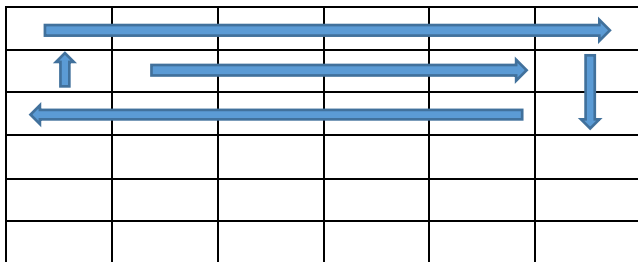
输出文件*helical.out*包括 N 行，每行 N 个正整数，每两个正整数之间用一个空格分开，方阵按行输出。

【输入输出样例】

<i>helical.in</i>	<i>helical.out</i>
6	1 2 3 4 5 6 14 15 16 17 18 7 13 12 11 10 9 8 13 12 11 10 9 8 14 15 16 17 18 7 1 2 3 4 5 6

【样例说明】

构造的方向可参照下图：



以上半部分为例。第一次指针向右移动到第1行第6格，分别填上1到6；第二次指针向下移动到第3行第6格，分别填上7和8；第三次指针向左移动到第3行第1格，分别填上9到13；第四次指针向上移动到第2行第1格，填上14；最后一次指针向右移动到第2行第5格，分别填上15到18。下半部分同理。

最终填好的方阵如下图：

1	2	3	4	5	6
14	15	16	17	18	7
13	12	11	10	9	8
13	12	11	10	9	8
14	15	16	17	18	7
1	2	3	4	5	6

【数据范围与测试点信息】

对于全部数据，满足 $N \leq 1000$ 。

对于测试点1－4，满足 $N \leq 20$ ；

对于测试点5－12，满足 $N \leq 200$ 。

5. 字符寻找 (*search.cpp*)

【题目背景】

对于反应力极差的 *QQYZ* 来说，在一个字符串中寻找一个特定的字符无疑是最难的事。然而由于他神奇的爱好，他又经常需要做这件事。

后来，当他学会了 C++ 语言（事实上他永远不可能学会）后，他才发现 C++ 语言是可以用来处理这个问题的。不过，他的 *OI* 水平导致他需要数个小时的时间才能处理完这个问题。

【题目描述】

QQYZ 给出了一个长度并不确定的字符串 *S*。同时，他又给出了一个字符 *ch*、一个长度为 2 的字符串 *T*，以及一个长度为 3 的字符串 *L*。

为了完成他的任务，他需要如下信息：

- 在字符串 *S* 中，字符 *ch* 出现的次数 x_1 ；
- 在字符串 *S* 中，字符串 *T* 出现的第一个位置 x_2 ；
- 在字符串 *S* 中，字符串 *L* 出现的最后一个位置 x_3 。

其中字符串 *T* 出现在字符串 *S* 中的定义为：字符串 *S* 存在至少一个长度和 *T* 相同的子串，且该子串和 *T* 完全相同；字符串 *T* 出现的位置就是该子串的起始位置。本题中字符串中的字符均从 1 开始编号，且区分大小写。

特殊地，我们规定：如果字符串 *T* 或 *L* 没有在 *S* 中出现，那么 x_2 或 x_3 的值就是 3334（3334 在试题中多次出现的原因是 *QQYZ* 非常喜欢看《三生三世十里桃花》，并且热衷于听其原声带歌曲）。

【输入格式】

输入文件 *search.in* 包括四行，第一行为字符串 *S*，第二行为字符 *ch*，第三行为字符串 *T*，第四行为字符串 *L*。

【输出格式】

输出文件 *search.out* 包括一行，共三个正整数 x_1, x_2, x_3 ，每两个正整数之间用一个空格隔开。

【输入输出样例】

<i>search.in</i>	<i>search.out</i>
3334slthqqyzqqbq q qq lth	5 9 6

【样例说明】

字符串 *S* 中出现了 5 个字符 '*q*'，故 $x_1 = 5$ ；第一次出现 "*qq*" 的位置为 9，故 $x_2 = 9$ ；最后一次出现 "*lth*"（也是唯一的一次）的位置为 6，故 $x_3 = 6$ 。本样例中没有出现字符串 *S* 不包含 *T* 或 *L* 的情况。

【数据范围与测试点信息】

对于全部数据，满足 $|S| \leq 10^5$ ，所有字符串中涉及的字符及字符 *ch* 均为可见符号、英文大小写字母、数字、空格之一。

对于测试点 1 – 3，满足字符串和字符 *ch* 均为英文大小写字母和数字；

对于测试点 4 – 6，满足字符串中没有空格，且字符 *ch* 非空格。

6. 智能标记 (*tab.cpp*)

【题目背景】

QQYZ 的英语翻译能力非常差，但是由于某种原因，他经常看英文书。但是他没有任何看懂5%以上书中单词的可能，所以他利用翻译笔这种设备帮自己翻译英文。每当翻译笔照出的光覆盖的书中的单词，*QQYZ* 就觉得非常开心。

但是，因为大小有限，翻译笔的翻译范围是有限的。在这种情况下，*QQYZ* 有必要统计接下来的单词中自己不认识的数量，再使用翻译笔，以提高自己阅读英文书籍的效率。

【题目描述】

我们假设 *QQYZ* 的阅读在同一页内进行。

已知 *QQYZ* 认识的单词有 N 个，分别为 s_1, s_2, \dots, s_N ；而本页内有 M 个单词，分别为 t_1, t_2, \dots, t_M 。*QQYZ* 知道翻译笔每次可以扫描连续的 K 个单词。

现在 *QQYZ* 可以选取 $M - K + 1$ 个位置为起点进行扫描。假设以第 i 个单词为起点所能扫描到的范围内有 cnt_i 个单词是 *QQYZ* 认识的，且 cnt_i 不低于一个给定的阈值 R ，那么 *QQYZ* 就要第 i 个单词到第 $i + K - 1$ 个单词中选出一个长度最长的单词（如果有多个长度最长的单词，就选择最靠前的一个），并统计该单词中给定字符 ch 的数量。

现在给出 N, M, K, R 的值，字符 ch 和字符串 $s_1, s_2, \dots, s_N, t_1, t_2, \dots, t_M$ ，请求出所有应统计单词中给定字符 ch 出现的数量之和。

【输入格式】

输入文件 *tab.in* 包括三行，第一行为四个正整数 N, M, K, R 和一个字符 ch ；第二行为 N 个字符串 s_1, s_2, \dots, s_N ；第三行为 M 个字符串 t_1, t_2, \dots, t_M 。每两个字符串中间用一个空格隔开。

【输出格式】

输出文件 *tab.out* 包括一行，共一个正整数，表示所有应统计单词中给定字符 ch 出现的数量之和。

【输入输出样例】

<i>tab.in</i>	<i>tab.out</i>
3 6 2 2 q qqyz qqbz slth qqbz slth qqyz qqyz qqbz qq	7

【样例说明】

t_1 和 t_2 中，有2个字符串被 *QQYZ* 认识，且最长的 t_1 中有3个 ' q '；

t_2 和 t_3 中，有2个字符串被 *QQYZ* 认识，且最长的 t_2 中不含 ' q '；

t_3 和 t_4 中，有2个字符串被 *QQYZ* 认识，且最长的 t_3 中有2个 ' q '；

t_4 和 t_5 中，有2个字符串被 *QQYZ* 认识，且最长的 t_4 中有2个 ' q '；

t_5 和 t_6 中，只有1个字符串被 *QQYZ* 认识，因此不进行统计。

所以应统计的 ' q ' 的数量之和为7。

【数据范围与测试点信息】

对于全部数据，满足 $N, K \leq M \leq 2000$ ， $|s_i|, |t_i| \leq 5000$ ，所有字符和字符串均为英文小写字母。

对于测试点1 - 12，满足 $N, M, K \leq 200$ ， $|s_i|, |t_i| \leq 100$ 。

7.素食主义者 (*vegetarian.cpp*)

【题目背景】

尽管 *QQYZ* 并不对此事十分有把握，这种人还是事实存在的：素食主义者。素食主义者分为严格素食主义者和一般素食主义者，他们的区别在于是否完全割断自己的生活与直接或间接使用动物材料的关系。

不过有一点是两种素食主义者都具备的：他们都爱吃素。因此面对一些菜肴时，他们会像小时候的 *QQYZ* 一样挑食（他现在已经不挑食了）。

【题目描述】

现在走廊上陈列了 N 种菜品，每种菜品有一个属性 k ， $k = 1$ 表示其为素食， $k = 2$ 表示其为荤食。有 M 个人相继走来，每个人也有一个属性 l ， $l = 1$ 表示其为素食主义者， $l = 2$ 则表示其不是。

每个人走来后，会进行以下三种操作之一：

- 形如 **G L R**，即将区间 $[L, R]$ 内的菜品取走。如果是素食主义者，则他不会取走荤食；已经被取走的菜品不会再取；

- 形如 **A**，即在当前所有菜品的末尾再增添一种属性与当前的人匹配的菜品，其中属性匹配是指素食主义者匹配素食，非素食主义者匹配荤食；

- 形如 **Q L R**，即询问区间 $[L, R]$ 内还剩下多少种属性与当前的人相匹配的菜品，属性匹配的定义与上一条操作相同。

特殊地，若取走操作和询问操作中 L, R 超出了区间 $[1, cnt]$ （ cnt 为当前菜品数量，已经被取走的也计算在内），则只对在区间内的部分进行处理。

现在给出 N 、 N 种菜品的属性以及 M 个操作，试对每次询问操作作出回答。

【输入格式】

输入文件 *vegetarian.in* 包括若干行，第一行为两个正整数 N, M ；第二行为 N 个正整数 k_i ；第三行为 M 个正整数 l_i ；接下来 M 行，每行有一种操作，每种操作由一个字母和若干个或 0 个正整数组成。

【输出格式】

输出文件 *vegetarian.out* 包括若干行，每行一个正整数，对每个询问操作作出回答，每次回答单独占一行。

【输入输出样例】

<i>vegetarian.in</i>	<i>vegetarian.out</i>
6 7	1
1 2 1 2 1 1	1
2 1 2 2 1 2 2	
G 1 2	
G 5 7	
A	
A	
Q 1 5	
G 5 8	
Q 2 6	

【样例说明】

下面我们用0,1表示一个菜品是否被取走：1为尚未被取走，0为已被取走， cnt 表示当前菜品数量（包括已经被取走的），则每次操作后状态如下：

• 初始状态（ $cnt = 6$ ）：

编号	1	2	3	4	5	6
属性	1	2	1	2	1	
状态	1					

• 第一次操作（ $cnt = 6$ ）：

非素食主义者，取走区间[1,2]内的菜品，因此可以取走菜品1,2；

编号	1	2	3	4	5	6
属性	1	2	1	2	1	
状态	0		1			

• 第二次操作（ $cnt = 6$ ）：

素食主义者，取走区间[5,7]内的菜品，等价于取走区间[5,6]内的菜品；由于菜品5,6均为素食，因此均可取走；

编号	1	2	3	4	5	6
属性	1	2	1	2	1	
状态	0		1		0	

• 第三次操作（ $cnt = 7$ ）：

非素食主义者，在后面加上一种荤食；

编号	1	2	3	4	5	6	7
属性	1	2	1	2	1		2
状态	0		1		0		1

• 第四次操作（ $cnt = 8$ ）：

非素食主义者，在后面加上一种荤食；

编号	1	2	3	4	5	6	7	8
属性	1	2	1	2	1		2	
状态	0		1		0		1	

• 第五次操作：

该素食主义者询问区间[1,5]内的素食数量，由上表可知答案为1；

• 第六次操作（ $cnt = 8$ ）：

非素食主义者，取走区间[5,8]内的菜品，因此可以取走菜品7,8；

编号	1	2	3	4	5	6	7	8
属性	1	2	1	2	1		2	
状态	0		1		0			

• 第七次操作：

该非素食主义者询问区间[2,6]内的荤食数量，由上表可知答案为1。

【数据范围与测试点信息】

对于全部数据，满足 $N, M \leq 2000$ ， $k_i, l_i \in \{1, 2\}$ 。

对于测试点1－4，满足不存在添加菜品的操作；

对于测试点5－8，满足不存在取走菜品的操作；

对于测试点9－12，满足 $k_i = l_i = 1$ 。

8. 拉链集合 (zip.cpp)

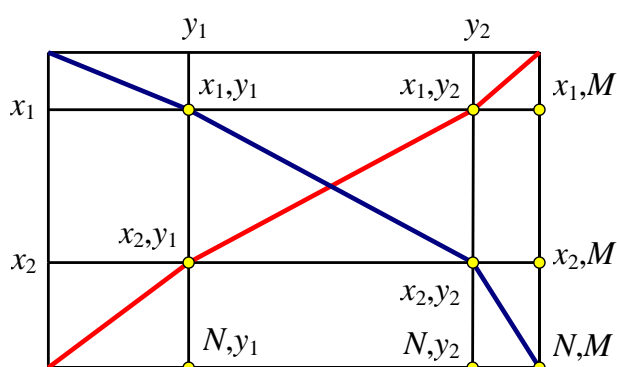
【题目背景】

QQYZ 喜欢穿带有拉链的衣物（利于在口袋里放置物品），但是由于他不爱护拉链，一个个拉链从衣物上脱落，被他收进了一个箱子。

这个箱子中的每个拉链具有不同的颜色，这导致 QQYZ 难以对它们进行分类。后来，他想到了一个好方法：把所有拉链排成一个矩阵，这样拉链就可以被鲜明地分成几个类别。

【题目描述】

已知 QQYZ 排出的矩阵有 N 行 M 列（拉链数量为 $N \cdot M$ ），他准备将这个矩阵分成 9 个小块。他要在矩阵内添加横向、纵向分界线各 2 条，这样矩阵就能被分成 3×3 个小块，如下图：



假设我们在第 x_1, x_2 行末尾添加两条横向分割线，在第 y_1, y_2 列末尾添加两条纵向分割线，并把左上角、右下角和中间三个矩形的主对角线和副对角线连起来（主对角线用蓝色表示，副对角线用红色表示），则我们可以得到两条折线。

如果一种分割方案使得两条折线均为直线，那么我们认为该分割方案是成功的。此时，我们需要求出中间那个矩形中所有拉链的颜色值之和。假设第 i 行第 j 列的拉链颜色值为 $col_{i,j}$ ，中间矩形的定义为左上角为 (x_1, y_1) 、右下角为 (x_2, y_2) 的矩形，即所有在 $(x_1, x_2]$ 行、 $(y_1, y_2]$ 列内的拉链。

给出 N, M 和所有矩阵中拉链的颜色值，求出所有成功的分割方案中中间矩形所有拉链的颜色值之和。

【输入格式】

输入文件 zip.in 包括若干行，第一行是两个正整数 N, M ；接下来 N 行，每行 M 个正整数，每两个正整数中间用一个空格隔开，表示 $col_{i,j}$ 。

【输出格式】

输出文件 zip.out 包括一行，共一个正整数，表示所有成功的分割方案中中间矩形所有拉链颜色值之和。由于答案可能很大，输出时请对 998244353 取模。如果没有任何一种分割方案是成功的，则输出 "Beautiful BYY"（不含引号）。

【输入输出样例1】

<i>zip.in</i>	<i>zip.out</i>
<pre> 6 6 1 3 4 2 5 3 5 3 3 2 6 1 6 4 3 1 5 2 3 2 1 4 5 4 2 4 5 5 2 6 4 5 2 3 1 6 </pre>	<pre> 64 </pre>

【样例说明】

选择 $x_1 = 1, x_2 = 5, y_1 = 1, y_2 = 5$ 和 $x_1 = 2, x_2 = 4, y_1 = 2, y_2 = 4$ 两种方案，均可满足题目要求。

第一种方案中间矩形为：

```

3 3 2 6
4 3 1 5
2 1 4 5
4 5 5 2

```

颜色值和为55；

第二种方案中间矩形为：

```

3 1
1 4

```

颜色值和为9，因此答案为64。

【输入输出样例2】

<i>zip.in</i>	<i>zip.out</i>
<pre> 4 5 3 5 4 7 2 4 4 2 4 3 3 1 3 2 5 5 4 2 7 6 </pre>	<pre> Beautiful BYY </pre>

【数据范围与测试点信息】

对于全部数据，满足 $N, M \leq 4000$ ， $col_{i,j} \leq 10^6$ 。

对于测试点1 – 6，满足 $N, M \leq 50$ ；

对于测试点7 – 12，满足 $\gcd(N - 1, M - 1) = 1$ ；

对于测试点13 – 18，满足 $\gcd(N - 1, M - 1) \leq 5$ 。