A. 二十四点

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

二十四点是一个非常有趣的游戏。游戏的目标是利用四则运算把几个数字给变成 24。

具体来说，游戏最开始会给出 n 个数字 $a_1, \ldots, a_n$。在游戏过程中，玩家每次可以选择两个下标不同（但是值可以相同）的数字 a_i, a_j，将 a_i, a_j 删除，并从 $a_i + a_j, a_i - a_j, a_i \times a_j, a_i / a_j$ 中选择一个加入到数列中（运算过程中允许小数）。如果在操作过程中，24 出现在了数列里，那么玩家胜利，否则玩家失败。

举例来说，如果初始数字是 $5, 5, 5, 5, 5$，一个可能的游戏过程是：

1. 删除最左侧两个 5，把 5×5 加入数列，得到 $5, 5, 5, 25$。
2. 删除最左侧两个 5，把 $5/5$ 加入数列，得到 $5, 25, 1$。
3. 删除 25 和 1，把 $25 - 1$ 加入数列，得到 $5, 24$。此时 24 出现在了数列里，玩家胜利。

注意这儿的规则和传统的 24 点不同，这儿允许有数字不被使用。

可怜定义一个数字序列 A 是好的当且仅当存在一个游戏过程能从 A 算出 24 点。举例来说 $2, 3, 4$ 就是好的，但 $1, 2, 3$ 不是。

现在可怜手上有 n 个数字 $x_1, \ldots, x_n$，可怜想要知道它 $2^n - 1$ 个非空子序列中，有多少个是好的。

Input

第一行输入一个整数 $n (1 \le n \le 10^5)$ 表示数字的个数。

第二行输入 n 个空格隔开的整数 $x_i (1 \le x_i \le 10)$，表示初始的数字。

Output

输出一行一个整数，表示答案，答案可能很大，对 998244353 取模后输出。

Example

input
6 1 2 3 4 5 6
output
32

B. 集合

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

最近放假了，可怜计划和她的好朋友 Sylvia 去公园玩。

我们可以把地图抽象成一个二维平面，可怜最开始的位置是点 x，Sylvia 最开始的位置是点 y，公园是一个以点 o 为圆心，半径为 r 的圆 O。

因为进入公园要买票，因此她们约好了先在不进入公园的情况下，在公园的边界上集合，然后在一起买票进公园。她们想要选择一个最优的集合地点使得双方要走的距离长度和最小。

换句话说，令 $d(s,t)$ 为在平面上，不经过圆 O 的内部，从 s 走到 t 的最短距离，可怜希望在圆 O 的边界上找到点 m，使得 $d(x,m)+d(y,m)$ 尽可能地小。

Input

输入第一行一个整数 $t(1 \le t \le 10^6)$ 表示数据组数。

对于每组数据，输入一行 7 个整数，分别是 x 的横纵坐标，y 的横纵坐标，o 的横纵坐标和半径 $r(1 \le r \le 10^3)$，其中所有坐标的绝对值不超过 10^3。

数据保证 x 和 y 都不在圆 O 的内部，即 x,y 和 o 的欧几里得距离都大于等于 r。

Output

对于每组数据，输出一行一个整数，表示最小的总距离，保留三位小数。

输入保证每组数据答案的第四位小数都不是 4 或者 5。

Example

input
3 0 1 2 0 1 0 1 0 0 2 0 3 0 1 0 0 2 0 1 3 1
output
2.571 2.000 4.472

C. 小游戏

time limit per test: 4.5 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

本题来源于《中国式家长》中的一个小游戏。

游戏开始时会生成一个长度为 n 的宝石序列，每个宝石按照从左到右的顺序被标号成 1 到 n。每个宝石有类型和能量两种属性，宝石的类型有四种，分别用字母ABCD表示，在初始时每个宝石的能量都是 1。同时保证初始时不存在连续的三个相同类型的宝石。

可怜可以进行若干轮操作，每轮操作的流程如下：

1. 选择结束游戏，当只剩下一个宝石的时候，可怜必须结束游戏。
2. 选择并删除某一个当前还存在的宝石，删除后右侧的宝石会自动向左补齐。

3. 令 i 为当前最小的下标满足从左到右第 $i, i+1, i+2$ 个宝石的类型和能量都相同。如果这样的 i 不存在，则这一轮结束，开始下一轮。否则这三个宝石会被移出游戏，同时在原来第 i 个宝石的位置会出现一个类型和原来相同，能量比原来大 1 的宝石。接着右侧的宝石会向左补齐，游戏回到这一步的开始，并继续从左到右寻找连续三个相同的宝石。

我们用 $A(x)$ 表示类型为 A 能量为 x 的宝石，下面是两个例子：

- 如果宝石序列是 A(1)A(1)B(1)A(1)A(1)，那么在删除 B(1) 之后，结果序列为A(2)A(1)。
- 如果宝石序列是 A(3)A(3)A(2)A(2)A(1)A(1)B(1)A(1)，那么在删除 B(1) 之后，结果序列为 A(4)。

可怜是一个全收集爱好者。她发现每一次开始这个小游戏，初始的宝石的数量、顺序和颜色都是完全相同的。而游戏中有一个传说级成就：合成出所有可能的结束状态。

在开始刷成就之前，可怜 想让你帮她计算一下，一共有多少种不同的结束状态。

两个结束状态是不同的当且仅当剩下宝石的数量不同，或者存在一个下标 i 满足两个状态中从左到右第 i 个小球的类型或者能量不同。

Input

第一行一个整数 $n(1 \le n \le 10^6)$ 表示宝石序列的长度。

第二行一个长度为 n 的由 ABCD 组成的字符串，表示初始的宝石序列。

Output

输出一行一个整数，表示可能的结束状态个数。答案可能很大，对 998244353 取模后输出。

Examples

input
3 ABA
output
6

input
5 AABAA
output
13

D. 精简改良

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

可怜最近在玩一款硬核战争游戏。

可怜控制的国家有 n 座城市，在 n 座城市之间有 m 条双向道路，第 i 条道路连接了城市 u_i 和 v_i，且通过这条道路的时间为 w_i。

最近，可怜点了名为"传送"的科技，这个科技可以让可怜的士兵能在国家的任意两个城市之间不经过道路快速地传送。这样从战争的角度来说，道路几乎就没有用了，可怜希望能删除一些道路，使得道路系统尽可能地不便，这样在敌人入侵时就能获得优势。

游戏的规则规定在任何时刻，同一个国家的任意两个城市之间都必须要能通过道路到达。因此可怜需要保证在删除道路之后，道路系统仍然是联通的。

令 $d(i,j)$ 为通过道路系统从第 i 个城市到第 j 个系统的最短时间，可怜定义一个道路系统的复杂度为 $\sum_{i=1}^{n}\sum_{j=i+1}^{n} d(i,j)$。可怜希望能删除一些道路，在保证图联通的情况下，使道路系统的复杂度尽可能的大。

Input

第一行输入两个整数 $n,m(1 \leq n \leq 14, 1 \leq m \leq \frac{n(n-1)}{2})$，表示城市数量与初始的道路数量。

接下来 m 行每行输入三个整数 $u_i,v_i,w_i(1 \leq u_i,v_i \leq n, 1 \leq w_i \leq 10^9)$，描述了一条道路。

输入保证图中没有自环、重边，同时图是联通的。

Output

输出一行一个整数，表示道路系统最大的可能的复杂度。

Examples

input
5 5 1 2 1 1 3 1 2 4 1 2 5 1 1 5 1
output
20

input
5 10 1 2 1 1 3 2 1 4 3 1 5 4 2 3 5 2 4 6 2 5 7 3 4 8 3 5 9 4 5 10
output
146

E. 最大独立集

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

树上最大独立集是个非常简单的问题，可怜想让它变得稍微难一点。

可怜最开始有一棵 n 个点无根树 T，令 $T(i)$ 为将点 i 作为根后得到的有根树。

可怜用 m 次操作构造了 $m+1$ 棵树 T'_0 至 T'_m，其中 $T'_0 = T(1)$。第 i 次操作，可怜选择了一个节点 k_i，它用如下的方式构造了 T'_i：

- 新建一棵和 $T(k_i)$ 一样的有根树 T_b。
- 新建 n 棵和 T'_{i-1} 一样的有根树，并将第 j 棵树的根节点的父亲设置为 T_b 中第 j 个点，即加上一条连接它们的边。
- 这样就得到了一个点数为 $\text{size}(T(k_i)) + n \times \text{size}(T'_{i-1})$ 的树 T_i，它的根节点是 T_b 中的节点 k_i。

现在可怜希望你对 T_0 至 T_m，分别求出这 $m+1$ 棵树的最大独立集大小。

有根树 T 的独立集 S 的定义是：S 是 T 上节点的子集，同时对于任意 S 中的点 i，i 的父亲 f_i 不在 S 中。

Input

第一行输入两个整数 $n, m (1 \le n, m \le 10^5)$，表示无根树上的节点个数与可怜进行的操作次数。

接下来 $n-1$ 行每行两个整数 $u, v (1 \le u, v \le n)$ 表示书上的一条边。

接下来 m 行每行一个整数 $k_i (1 \le k_i \le n)$ 表示在第 i 次操作中，可怜选择的节点。

Output

输出 $m+1$ 行，每行一个整数，表示对应的树的最大独立集的大小。答案可能很大， 对 998244353 取模后输出。

Examples

input
1 5 1 1 1 1 1
output
1 1 2 2 3 3

input

```
5 5
1 2
2 3
2 4
1 5
5
4
3
2
1
```

output

```
3
18
93
465
2328
11643
```

F. 小清新数论

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

这是一道比较基础的数论题。

给出一个整数 n，计算 $\sum_{i=1}^{n}\sum_{j=1}^{n}\mu(\gcd(i,j))$。

其中 $\gcd(i,j)$ 表示 i,j 的最大公约数，$\mu(i)$ 表示莫比乌斯函数，它的一个等价定义是 $\mu(1)=1$，$\mu(n)=-\sum_{d<n,d|n}\mu(d)$ 当 $n>1$ 时。

Input

输入一行包含一个整数 $n(1\le n\le 10^{10})$。

Output

输出一行一个整数，表示答案。答案可能很大，请对 998244353 取模后输出。

Examples

input
5
output
14

input
100
output
3631

G. 排列

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

说出来你可能不信，这场比赛居然有签到题。

定义长度为 n 的排列 p 的前缀最小值数组为一个长度为 n 的数组 $A(p)$，其中 $A(p)_i = \min_{j=1}^{i} p_j$。

可怜定义排列 p 的长度为 i 的前缀小于长度为 j 的前缀当且仅当 $A(p)_i < A(p)_j$ 或者 $A(p)_i = A(p)_j$ 且 $i < j$。利用这个小于关系，可怜求得了另一个长度为 n 的排列 q，其中 q_i 表示排列 p 第 i 小的前缀的长度。

可怜是个粗心的女孩子，因为一些机缘巧合，可怜丢失了排列 p。于是可怜希望能够通过排列 q 来还原出排列 p。满足条件的排列 p 可能有很多，可怜希望你能求出它们中字典序最小的那个。

长度为 n 的排列 x 在字典序上小于长度为 n 的排列 y 当且仅当存在下标 $i \in [1, n]$ 满足 $x_i < y_i$ 且对更小的下标 $j \in [1, i)$，都有 $x_i = y_i$。

Input

第一行输入一个整数 $n(1 \le n \le 10^5)$，表示排列的长度。

第二行输入一个长度为 n 的排列 q。

输入保证存在至少一个满足条件的排列 p。

Output

输出一行一个长度为 n 的排列，表示字典序最小的满足条件的排列 p。

Example

input
5 5 3 4 1 2
output
3 4 2 5 1

H. 涂鸦

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

九条可怜收到了一块神奇的画板。

画板被划分成了 n 行 m 列一共 $n \times m$ 个格子，记第 i 行第 j 列的格子的坐标为 (i, j)。画板的每一行有属性 $l_i, r_i (1 \le l_i \le r_i \le m)$，代表了 $[1, m]$ 的一个子区间。

画板有一个开始按钮，在按下开始按钮之后，画板会自动初始化颜色：对于第 i 行，画板会从 $[l_i, r_i]$ 的 $\frac{(r_i - l_i + 1)(r_i - l_i + 2)}{2}$ 个子区间中，独立地等概率地选择一个涂黑。举例来说，如果 $i = 1, [l_i, r_i] = [1, 2]$，那么它

一共有 3 个子区间 $[1,2],[1,1]$ 和 $[2,2]$，他们各有 $\frac{1}{3}$ 的概率被选中，因此格子 $(1,1),(1,2)$ 都有 $\frac{1}{2}$的概率被染黑。

在画板初始化结束后，可怜选择了 q 个矩形，第 i 个矩形包含了所有满足 $a \in [lx_i, rx_i], b \in [ly_i, ry_i]$ 的格子 (a,b)。接着可怜会把至少被一个矩形包含的格子给涂白（如果原来就是白色则颜色不变）。

可怜重复上述过程若干次，渐渐地感到了无聊。为了让事情更有趣一些，她想要计算，在给出这 q 个矩形的情况下，绘画结束后画板上黑色格子数量的期望是多少。

Input

第一行三个整数 $n, m, q (1 \leq n, q \leq 2 \times 10^5, 1 \leq m \leq 10^9)$。

接下来 n 行每行两个整数 $l_i, r_i (1 \leq l_i \leq r_i \leq m)$，表示第 i 行的属性。

接下来 q 行每行四个整数 $lx_i, ly_i, rx_i, ry_i (1 \leq lx_i \leq rx_i \leq n, 1 \leq ly_i \leq ry_i \leq m)$，表示一个矩形。

Output

输出一行一个整数表示答案，棋盘上黑色格子数量的期望对 998244353 取模后的结果。即，如果答案的最简分数表示为 $\frac{x}{y}$，输出 $x \times y^{-1} \mod 998244353$。

Example

input
4 4 1 1 3 2 4 3 4 1 4 2 2 3 3
output
831870299

I. 石头剪刀布

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

可怜去观看了石头剪刀布的世界最高赛事 WRSP。

今年的比赛一共有 n 名选手参加，在比赛开始时，每名选手都会收到一张卡片，这张卡片上写着剪刀、石头、布中的一个。显然初始的卡牌分配情况有 3^n 种。

比赛场地一共有 n 个座位，最开始第 i 个选手坐在第 i 个座位上。

接下来发生了 m 个事件，事件有两种：

- $1\ x\ y$，主办方撤去了第 y 个座位，原来在第 y 个座位上的选手 b 需要和 x 个座位上的选手 a 利用他们的卡片进行一场石头剪刀布比赛，如果 b 赢了 a，则选手 a 被淘汰，选手 b 坐到第 x 个座位上；否则（打平或者 b 输了），则选手 b 被淘汰，选手 a 的坐位不变。

- $2\ x$，可怜提出了一个问题，她想要知道在进行了之前的所有第 1 类事件后，有多少种卡牌分配情况可以让第 x 个选手到现在还没有被淘汰。

Input

第一行输入两个整数 $n, m(1 \leq n, m \leq 2 \times 10^5)$，表示选手个数和事件个数。

接下来 m 行，每行描述了一个事件。如果是第一类事件，则输入三个整数 $1\ x\ y(1 \leq x, y \leq n, x \neq y)$ 且这两个座位在之前没有被撤去；如果是第二类事件，则输入两个整数 $2\ x(1 \leq x \leq n)$。

Output

对于每个第二类事件，输出一行一个整数，表示这个选手还没有被淘汰的分配情况个数对 998244353 取模后的值。

Example

input
3 5 2 1 1 2 1 2 1 1 2 3 2 1
output
27 9 6

J. 子序列

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

可怜发现了一种快速构造超长字符串的方法。

最开始，可怜手上有一个空串 s_0，接着，可怜对这个串进行了 n 次操作。在第 i 次操作的时候，可怜选择了一个字符 c，并利用 s_{i-1} 和 c 构造了字符串 $s_i = cs_{i-1}[1]cs_{i-1}[2] \dots s_{i-1}[m]c$，其中 m 为串 s_{i-1} 的长度。

举例来说，如果 $n = 3$ 且可怜选择的字符分别是 a, b, c，则 $s_n = cbcacbc$。

现在，可怜想要知道，字符串 s_n 本质不同的非空子序列有多少个。（注意不能为空，但可能是 s_n 本身）。

串 s 是串 t 的子序列当且仅当串 s 可以通过删去 t 中的某些字符得到。

Input

第一行输入一个整数 $n(1 \leq n \leq 2000)$，表示可怜进行的操作次数。

第二行输入一个长度为 n 的小写字符串 s，其中 s_i 表示可怜在第 i 轮选择的字符。

Output

输出一行一个答案，表示 s_n 中本质不同的子序列个数，答案可能很大，对 998244353 取模后输出。

Examples

input
2 ab
output
6

input
10 aaaaaaaaaa
output
1023