

# 浙江理工大学 ACM 集训队

## 7 月训练赛

### 试题册

Hangzhou, 2026.7.9

### 试题列表

| 编号 | 题目名称            |
|----|-----------------|
| A  | LittleQiu 的计组作业 |
| B  | LittleQiu 的编程作业 |
| C  | LittleQiu 的英语作业 |
| D  | LittleQiu 的高数作业 |
| E  | LittleQiu 的离散作业 |
| F  | LittleQiu 的线代作业 |
| G  | LittleQiu 的化学作业 |
| H  | LittleQiu 的大物作业 |
| I  | LittleQiu 的计组课设 |
| J  | LittleQiu 的生物作业 |
| K  | LittleQiu 的历史作业 |
| L  | LittleQiu 的地理作业 |

本试题册共 12 题，共 21 页。

## Problem A. LittleQiu 的计组作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 2 s  
内存限制 512 MB

LittleQiu 收到了他的计组作业，作业内容如下：

在某个 RISC 处理器的设计中，微码控制器使用 16 条基本微指令，分别编号为 0 到 15。每条微指令执行时会消耗一定的动态能耗，恰好等于其编号值。

一个微程序是由若干条互不重复的微指令组成的序列。微程序的总能耗等于其中所有微指令编号之和。微程序可以按以下规则进行全序比较：

1. **长度优先**：微程序长度（指令条数）短的更小；
2. **字典序**：若长度相同，则按指令编号的字典序比较，其中编号 0 ~ 9 对应字符 '0' ~ '9'，编号 10 ~ 15 对应字符 'A' ~ 'F'（比较时使用 ASCII 顺序，即 '0' < '1' < ... < '9' < 'A' < ... < 'F'）。

例如，能耗为 20 的微程序有 "5F"（ $5 + 15 = 20$ ）、"6E"（ $6 + 14 = 20$ ）等，按上述规则，最小的程序是 "5F"（长度 2，且 '5' 最小）。

现在，处理器设计者希望你能自动生成满足指定能耗的微程序。

给定目标能耗  $n$  和一个正整数  $k$ ，请输出满足能耗为  $n$  的第  $k$  小的微程序（1-indexed）。如果不存在（即总数量小于  $k$ ），输出 -1。

注意：在本题中，需要考虑同一程序中指令的顺序，即 5E 和 E5 视为两种不同的程序。

### 输入格式

第一行一个整数  $T$ （ $1 \leq T \leq 10000$ ）。

接下来  $T$  行，每行两个整数  $n, k$ （ $0 \leq n \leq 120$ ， $1 \leq k \leq 10^9$ ）。

### 输出格式

对于每组询问，输出一行一个字符串，表示第  $k$  小的微程序，或 -1。

### 样例

| 标准输入  | 标准输出 |
|-------|------|
| 6     | 0    |
| 0 1   | 5F   |
| 20 1  | 7D   |
| 20 3  | F5   |
| 20 10 | 3DEF |
| 45 1  | -1   |
| 0 2   |      |

## 提示

- $n = 0, k = 1$ : 能耗为 0 的微程序只有 "0", 第 1 小是 "0"。
- $n = 20$ : 所有能耗为 20 的程序按序为 "5F", "6E", "7D", "8C", "9B", "B9", "C8", "D7", "E6", "F5" (顺序有意义, 如 "E6" 与 "6E" 是两个不同的程序)。故第 1 小是 "5F", 第 3 小是 "7D", 第 10 小是 "F5"。
- $n = 45, k = 1$ : 最短程序长度为 4, 字典序最小的是 "3DEF" ( $3 + 13 + 14 + 15 = 45$ )。
- $n = 0, k = 2$ : 能耗为 0 的微程序总共只有 1 个, 不存在第 2 小, 输出 -1。

## Problem B. LittleQiu 的编程作业

|      |        |
|------|--------|
| 输入文件 | 标准输入   |
| 输出文件 | 标准输出   |
| 时间限制 | 1 s    |
| 内存限制 | 256 MB |

LittleQiu 收到了他的编程作业，作业内容如下：

在编译器后端优化中，**常量传播**（Constant Propagation）是一个基础而强大的优化手段。它通过数据流分析，追踪程序中每个变量在编译时是否为常量，并将变量引用替换为常量值，从而为后续的死代码消除、循环展开等优化提供机会。

本题模拟一个「理想化」的编译器优化器。给定一个程序的源代码（以抽象语法树的序列化文本形式描述），你的任务是执行以下完整流程：

1. 构建控制流图（CFG），并计算每个基本块的**到达定值**（Reaching Definitions）集合。
2. 利用**迭代数据流算法**，求解每个变量的**常量值格**（Constant Lattice），其中格值包括  $\perp$ （未定义）、 $c$ （具体常量）、 $\top$ （非常量）。
3. 结合**支配树**（Dominator Tree）和**逆后序**（Reverse Postorder）遍历，加速不动点求解。
4. 最后，将程序中所有 `print` 或 `output` 语句的参数替换为常量值，得到最终输出字符串。

然而，经过对数百个开源项目（包括 Linux 内核、LLVM、WebKit 等）的统计，我们发现所有程序在「理想化」条件下最终输出的字符串都惊人地一致。这是因为我们的测试数据集均来自一个神秘的代码仓库，该仓库中的所有程序都经过一个「终极归一化」处理，使得它们无论执行何种计算，最终的输出结果总是 `Hello_World`。

### 输入格式

第一行一个整数  $L$  ( $0 \leq L \leq 10^6$ )，表示程序文本的行数。

接下来  $L$  行，为待优化程序的抽象语法树序列化结果（每行可以为空）。

保证输入总大小不超过 1MB ( $10^6$  字节)，仅包含可见 ASCII 字符、空格与换行。

### 输出格式

输出一行，为该程序经「理想化」优化后的最终输出字符串。

## 样例

| 标准输入                                                                             | 标准输出                   |
|----------------------------------------------------------------------------------|------------------------|
| <pre>5 func main() {     x = "Hello";     y="World";     print(x+"_"+y); }</pre> | <pre>Hello_World</pre> |

## Problem C. LittleQiu 的英语作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 1 s  
内存限制 256 MB

LittleQiu 收到了他的英语作业，作业内容如下：

你需要抄写一段作文范文，记为字符串  $s_1$ 。但是由于 LittleQiu 非常自恋，他想要在这个字符串中找到一处地方插入自己的座右铭  $s_2$ 。为了不被发现，他希望  $s_2$  有尽可能长的前缀与插入点之后的  $s_1$  原文重合。请你帮他找到这个位置。

形式化地：字符串下标从 0 开始。选择一个插入位置  $p$  ( $0 \leq p \leq |s_1|$ )，表示把  $s_2$  插入在  $s_1$  的第  $p$  个字符之前 ( $p = |s_1|$  表示插入在末尾)。该位置的**重合长度**定义为  $s_2$  与  $s_1$  的后缀  $s_1[p..|s_1|-1]$  的最长公共前缀长度。你要求出使重合长度最大的  $p$ ；若有多个  $p$  的重合长度相同且最大，输出其中最小的。

例如，要在 I like this little aplle. 中插入 littleqiu，最佳位置为 12（即字母 l，此处重合长度为 6，重合部分为 little）。

### 输入格式

第一行一个字符串  $s_1$  ( $1 \leq |s_1| \leq 10^5$ )。

第二行一个字符串  $s_2$  ( $1 \leq |s_2| \leq 10^5$ )。

字符串可能包含大小写字母、数字、英文符号（, . ! ? '）和空格（空格可能出现在字符串中间任意位置，行首行尾不会出现空格）。

### 输出格式

一行一个整数，表示最适合插入座右铭的位置  $p$ 。

### 样例

| 标准输入                                   | 标准输出 |
|----------------------------------------|------|
| I like this little aplle.<br>littleqiu | 12   |

### 提示

$s_1[12..]$  为 little aplle.，与 littleqiu 的最长公共前缀为 little，长度 6；其余位置的重合长度均小于 6，故答案为 12。

## Problem D. LittleQiu 的高数作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 2 s  
内存限制 256 MB

这是一道交互题。

LittleQiu 收到了他的高数作业，作业内容如下：

有一个函数  $f(x)$ ，已知其定义域为  $[l, r]$ ，且是关于  $x$  的严格单峰函数（存在唯一极大值点  $x_0 \in [l, r]$ ， $f$  在  $[l, x_0]$  上严格递增、在  $[x_0, r]$  上严格递减； $x_0$  可能位于区间端点）。

你可以反复询问  $f(x)$  在某一特定点的取值，请求出它的极值点  $x_0$ 。

### 交互

交互开始时，评测机首先输出一行两个实数  $l, r$  ( $-10^5 \leq l \leq r \leq 10^5$ )，表示  $f$  的定义域。保证  $l \leq x_0 \leq r$ ，且对任意询问点  $|f(x)| \leq 10^{12}$ 。

接下来你可以进行至多 100 次询问。每次询问输出一行：

1. ? x: 其中  $x$  为实数且必须满足  $l \leq x \leq r$ 。评测机将回答一行一个实数，表示  $f(x)$  的值（不超过 17 位有效数字的科学计数法或小数表示）。
2. ! x: 其中  $x$  为你找到的极值点。输出后你的程序应立即正常退出。此操作不计入询问次数，且必须恰好进行一次。

每次输出后都必须刷新缓冲区（C++ 使用 `cout.flush()` 或 `endl`，C 使用 `fflush(stdout)`，Python 使用 `flush=True`）。

当你的答案  $x$  与标准答案  $x_0$  满足

$$\min \left( |x - x_0|, \left| \frac{x - x_0}{x_0} \right| \right) \leq 10^{-3}$$

（绝对误差或相对误差不超过  $10^{-3}$ ）时，认为你的答案正确。

询问超过 100 次、询问点越出定义域或输出格式非法均被判为 Wrong Answer。

### 样例

| 标准输入 | 标准输出     |
|------|----------|
| 0 10 | ? 2      |
| -9   | ? 5      |
| 0    | ? 8      |
| -9   | ! 5.0002 |

## 提示

交互样例中  $f(x) = -(x-5)^2$ : 依次询问  $f(2) = -9$ 、 $f(5) = 0$ 、 $f(8) = -9$ , 最后回答 5.0002, 因  $|5.0002 - 5| \leq 10^{-3}$  被判为正确。样例仅用于说明格式, 实际询问次数与策略由你决定。



## Problem E. LittleQiu 的离散作业

|      |        |
|------|--------|
| 输入文件 | 标准输入   |
| 输出文件 | 标准输出   |
| 时间限制 | 2 s    |
| 内存限制 | 512 MB |

LittleQiu 收到了他的离散作业，作业内容如下：

你有一种特别的逻辑表达式。在这种逻辑表达式中，所有操作数都是变量，且它们的取值只能为 0 或 1，运算从左往右进行。如果表达式中有括号，则先计算括号内的子表达式的值。特别的，这种表达式有且仅有以下几种运算：

1. 与运算： $a \& b$ 。当且仅当  $a$  和  $b$  的值都为 1 时，该表达式的值为 1。其余情况该表达式的值为 0。
2. 或运算： $a | b$ 。当且仅当  $a$  和  $b$  的值都为 0 时，该表达式的值为 0。其余情况该表达式的值为 1。
3. 取反运算： $!a$ 。当且仅当  $a$  的值为 0 时，该表达式的值为 1。其余情况该表达式的值为 0。

现在给定一个逻辑表达式和其中每一个操作数的初始取值，对每次询问，回答：将某一个操作数的值取反后，整个表达式的值是多少。

为了化简对表达式的处理，我们有如下约定：

表达式将采用**后缀表达式**的方式输入。后缀表达式的定义如下：

1. 如果  $E$  是一个操作数，则  $E$  的后缀表达式是它本身。
2. 如果  $E$  是  $E_1 \text{ op } E_2$  形式的表达式，其中  $\text{op}$  是任何二元操作符，且优先级不高于  $E_1$ 、 $E_2$  中括号外的操作符，则  $E$  的后缀式为  $E'_1 E'_2 \text{ op}$ ，其中  $E'_1$ 、 $E'_2$  分别为  $E_1$ 、 $E_2$  的后缀式。
3. 如果  $E$  是  $(E_1)$  形式的表达式，则  $E_1$  的后缀式就是  $E$  的后缀式。

同时为了方便，输入中：

1. 与运算符 ( $\&$ )、或运算符 ( $|$ )、取反运算符 ( $!$ ) 的左右均有一个空格，但表达式末尾没有空格。
2. 操作数由小写字母  $x$  与一个正整数拼接而成，正整数表示这个变量的下标。例如  $x_{10}$  表示下标为 10 的变量  $x_{10}$ 。数据保证每个变量在表达式中出现恰好一次。

## 输入格式

第一行包含一个字符串  $s$  ( $1 \leq |s| \leq 10^6$ )，表示上文描述的表达式。

第二行包含一个正整数  $n$  ( $1 \leq n \leq 10^5$ ), 表示表达式中变量的数量。表达式中变量的下标为  $1, 2, \dots, n$ 。

第三行包含  $n$  个整数, 第  $i$  个整数表示变量  $x_i$  的初值。

第四行包含一个正整数  $q$  ( $1 \leq q \leq 10^5$ ), 表示询问的个数。

接下来  $q$  行, 每行一个正整数, 表示需要取反的变量的下标。注意, 每一个询问的修改都是临时的, 即之前询问中的修改不会对后续的询问造成影响。

数据保证输入的表达式合法, 变量的初值为 0 或 1。

## 输出格式

输出共  $q$  行, 每行一个 0 或 1, 表示该询问下表达式的值。

## 样例

| 标准输入       | 标准输出 |
|------------|------|
| x1 x2 & x3 | 1    |
| 3          | 1    |
| 1 0 1      | 0    |
| 3          |      |
| 1          |      |
| 2          |      |
| 3          |      |

## 提示

原表达式为  $(x_1 \& x_2) | x_3$ 。

- 取反  $x_1$ :  $(0 \& 0) | 1 = 1$ ;
- 取反  $x_2$ :  $(1 \& 1) | 1 = 1$ ;
- 取反  $x_3$ :  $(1 \& 0) | 0 = 0$ 。

## Problem F. LittleQiu 的线代作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 1 s  
内存限制 256 MB

LittleQiu 收到了他的线代作业，作业内容如下：

在一个  $n$  维空间的球面上，有  $n+1$  个点。保证这些点处于一般位置（任意两点不重合、三点不共线、四点不共面，以此类推），因此过这  $n+1$  个点的球面唯一存在。

现在，给你这些点的坐标，请你求出球心的坐标。

### 输入格式

第一行一个整数  $n$  ( $2 \leq n \leq 15$ )，表示维度。

接下来  $n+1$  行，每行  $n$  个实数（最多 6 位小数，绝对值不大于 20000），表示一个点的坐标。

数据保证方程组条件良好。

### 输出格式

一行  $n$  个实数，用空格隔开，表示你求得的球心坐标，保留三位小数。

本题采用 Special Judge：当你输出的球心某一维  $x$  与标准答案对应维  $x_0$  满足

$$\min \left( |x - x_0|, \left| \frac{x - x_0}{x_0} \right| \right) \leq 0.01$$

（绝对误差或相对误差不超过 0.01）时，认为该维正确；每一维都正确即判为通过。

### 样例

| 标准输入                                | 标准输出        |
|-------------------------------------|-------------|
| 2<br>0.0 0.0<br>-1.0 1.0<br>1.0 0.0 | 0.500 1.500 |

### 提示

点  $(0,0)$ 、 $(-1,1)$ 、 $(1,0)$  到  $(0.5,1.5)$  的距离均为  $\sqrt{2.5}$ ，故球心为  $(0.500, 1.500)$ 。

## Problem G. LittleQiu 的化学作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 3 s  
内存限制 512 MB

LittleQiu 收到了他的化学作业，作业内容如下：

给出一个有机化学式（一定含 C H，可能含 O N），你需要判断存不存在这个有机物（C 为四价，H 为一价，O 为二价，N 为三价），并构造一个可行的结构式（即表示 C 的节点的度为 4，表示 H 的节点的度为 1，表示 O 的节点的度为 2，表示 N 的节点的度为 3，其中一条  $k$  重键为两端节点的度各贡献  $k$ ，整个分子必须连通）。

### 输入格式

输入一个字符串，代表这个有机化学式。

保证化学式一定含 C、H（可能含 O、N），且有机物含原子个数小于  $2 \times 10^6$ 。

### 输出格式

若不存在这个有机物，输出 0。

否则第一行输出两个数  $n, m$ ，分别表示构造的图的点数和边数。

第二行输出  $n$  个符号，第  $i$  个表示点  $i$  的原子符号。

接下来输出  $m$  行，每行三个数，前两个表示这两个原子的编号，第三个表示这条键有几重。  
不能输出已经输出过的键。

存在多解时输出任意一组即可。

### 样例

| 标准输入 | 标准输出                                                            |
|------|-----------------------------------------------------------------|
| CH4O | 6 5<br>C H H H H O<br>1 2 1<br>1 3 1<br>1 4 1<br>1 6 1<br>6 5 1 |

| 标准输入   | 标准输出                                                                                                                                                                                  |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| C7H6O2 | 15 15<br>C C C C C C C H H H H H H O O<br>1 2 3<br>2 3 1<br>3 4 2<br>4 5 1<br>5 6 1<br>6 7 2<br>7 1 1<br>3 8 1<br>4 9 1<br>5 14 1<br>14 10 1<br>5 15 1<br>15 11 1<br>6 12 1<br>7 13 1 |

## 提示

样例 1 构造的是甲醇  $\text{CH}_3\text{OH}$ ：碳连三个氢和一个氧，氧再连一个氢，全部为单键。样例 2 给出的是  $\text{C}_7\text{H}_6\text{O}_2$  在本题（图论）意义下的一种合法结构；合法结构不唯一，输出任意一种即可。

## Problem H. LittleQiu 的大物作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 2 s  
内存限制 512 MB

LittleQiu 收到了他的大物作业，作业内容如下：

某城市新建了  $n$  个居民小区，每个小区位于一个整数坐标点  $(x_i, y_i)$  上。

市政公司计划铺设若干条**水平或垂直**的供暖管道来连接所有小区，使得任意两个小区之间都能通过管道网络相通。管道只能在同一水平线或同一垂直线上铺设，并且两小区之间若铺设管道，其长度（曼哈顿距离）不能超过某个阈值  $T$  米。

由于预算紧张，市政公司还**允许额外修建一个临时换热站**，可以放置在任意整数坐标点（不能与已有小区重合），该换热站也必须通过管道与所有小区连通。

请你帮助市政公司计算：**最小的整数  $T$** ，使得存在一种换热站建设方案（也可以不建）和管道铺设方案，满足所有要求。如果不存在这样的  $T$ ，输出  $-1$ 。

曼哈顿距离：两点  $(x_1, y_1)$  与  $(x_2, y_2)$  之间的曼哈顿距离为  $|x_1 - x_2| + |y_1 - y_2|$ 。

注：在本题中，管道只要求水平或垂直，允许两小区之间使用一段折线形状的管道连接。

### 输入格式

第一行包含一个整数  $n$  ( $2 \leq n \leq 1000$ )。

接下来  $n$  行，每行两个整数  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^9$ )，表示第  $i$  个小区的坐标。保证所有点互不相同。

### 输出格式

输出一个整数，表示最小的可行  $T$ 。

### 样例

| 标准输入                                    | 标准输出 |
|-----------------------------------------|------|
| 4<br>100 0<br>0 100<br>-100 0<br>0 -100 | 100  |
| 标准输入                                    | 标准输出 |
| 2<br>0 0<br>1 2                         | 2    |

## 提示

**样例 1:** 可以在原点  $(0,0)$  修建换热站。四个小区到原点的曼哈顿距离均为 100，每条管道长度均为 100，网络连通。可以证明  $T < 100$  时无法连通。

**样例 2:** 两个小区的曼哈顿距离为 3，直接相连需要  $T = 3$ ；在  $(0,1)$  修建换热站后，它到两个小区的距离分别为 1 和 2，故  $T = 2$  即可。可以证明  $T = 1$  不可行。

## Problem I. LittleQiu 的计组课设

输入文件 标准输入  
输出文件 标准输出  
时间限制 2 s  
内存限制 512 MB

本题为提交答案题（程序只需按任务编号原样打印你预先构造的指令序列）。

LittleQiu 收到了他的计组课设，课设内容如下：

你初始拥有四个 01 栈，分别命名为 A, B, C, D，四个栈中都被预先压入  $10^{10^{10}}$  个 0。然后，老师向 A, B 两栈中分别压入了一个 64 位无符号整数（高位在下、低位在上，不足 64 位则用前导零补齐 64 位），你需要使用下面指令表中的指令，从 10 个任务中任选至少 3 个完成。

### 指令表

- 与非运算（NAND）：NAND X Y Z 表示先弹出 X 的栈顶元素  $x$ ，再弹出 Y 的栈顶元素  $y$ （不放回），将  $\neg(x \wedge y)$  压入 Z 栈。  
补充：形如 NAND A B B、NAND A A B 的操作也是被允许的，其中 NAND A A B 弹出的是 A 的栈顶两个元素，而不是同一个元素两次。
- 复制（DUP）：DUP X 将 X 栈顶元素复制一份重新压入 X。
- 删除（POP）：POP X 删除 X 栈顶元素。
- 结束标记（END）：标记程序结束。每段程序必须以恰好一个 END 结尾。

### 二进制编码

本题采用 8 位二进制指令编码：

| 含义     | 操作符 | 栈 1 | 栈 2 | 栈 3 |
|--------|-----|-----|-----|-----|
| 所在二进制位 | 1,2 | 3,4 | 5,6 | 7,8 |

栈 A, B, C, D 分别用 00, 01, 10, 11 表示；操作符 NAND, DUP, POP, END 分别用 00, 01, 10, 11 表示。指令不需要的字段可用任意 01 补齐。每条指令得到 8 位后，以两位大写十六进制输出，指令之间用一个空格分隔。

例如（x 为任意 01）：

- NAND A B C 编码为 00000110，十六进制 06；
- NAND D D C 编码为 00111110，十六进制 3E；
- DUP A 编码为 0100xxxx，x 全取 0 时为 40；
- POP A 编码为 1000xxxx，x 全取 0 时为 80；



5. END 编码为 11xxxxxx, x 全取 0 时为 C0。

### 任务表

以下十个任务，正确完成其中至少 3 个即可通过本题（A, B 分别代表压入对应栈的数；要求结果的存储顺序与题目中 A, B 的存储方式相同，即高位在下、低位在上）：

1. 将  $A + B$  的值存到 C 中（对  $2^{64}$  取模）。
2. 将  $A \times B$  的值存到 C 中（对  $2^{64}$  取模）。
3. 将  $A \oplus B$ （按位异或）的值存到 C 中。
4. 将  $A - B$  的值存到 C 中（对  $2^{64}$  取模）。
5. 对每一位  $i$ ：若  $A_i$  为 0，则  $C_i = B_i, D_i = 0$ ；否则  $C_i = 0, D_i = B_i$ 。
6. 将 A, B 视为 64 位有符号整数（补码），将  $A + B$  存到 C 中。
7. 将 A, B 视为 64 位有符号整数（补码），将  $A \times B$ （的低 64 位）存到 C 中。
8. 分别统计 A 和 B 中所有 1 的个数并求和，将总个数作为 64 位数存入 C 中（低位在栈顶，高位补 0）。
9. 将 A 中的 64 位数循环左移  $B \bmod 64$  位（高位循环到低位），结果存入 C 中。
10. 将 A, B 中较大的数（按无符号比较）存入 C 中。

## 输入格式

输入共 10 行，每行一个整数 sid（构成 1 ~ 10 的一个排列），表示需要回答的任务编号。

## 输出格式

对每行读入的 sid，输出一行对应任务的指令序列（十六进制、空格分隔）。若你未选择完成该任务，输出单独一个 END 指令（如 C0）即可。

每行程序至多  $10^5$  条指令，且必须以 END 结尾，否则直接判为 Wrong Answer。

**评测方式 (Special Judge)：**对你输出的每行程序，评测器会在 60 组数据（12 组边界值 + 48 组随机数，随机种子由该测试点的任务排列决定）上模拟栈机执行：向 A, B 压入两个 64 位数并运行你的程序，然后检查 C 栈（任务 5 还检查 D 栈）栈顶 64 个元素（栈顶为最低位）是否等于期望结果；其余栈的最终内容不作要求。全部 60 组数据都正确的任务视为「完成」。完成的任务数  $\geq 3$  即通过该测试点，否则判为 Wrong Answer。

## 样例

| 标准输入 | 标准输出                    |
|------|-------------------------|
| 1    | 40 50 06 C0             |
| 2    | C0                      |
| 3    | 40 50 06 60 08 19 06 C0 |
| 4    | C0                      |
| 5    | C0                      |
| 6    | C0                      |
| 7    | C0                      |
| 8    | C0                      |
| 9    | C0                      |
| 10   | C0                      |

## 提示

样例仅示意输入输出格式，输出内容并非任何任务的正确解。

## Problem J. LittleQiu 的生物作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 2 s  
内存限制 256 MB

LittleQiu 收到了他的生物作业，作业内容如下：

某种性状由  $m$  个相互独立的基因位点共同决定。每个位点有一对等位基因，显性记为大写、隐性记为小写。

现有一对夫妇，他们在这  $m$  个位点上**全部**都是杂合子（即每个位点的基因型都是  $Aa$  形式）。根据孟德尔遗传定律， $Aa \times Aa$  杂交的后代在该位点上为**显性纯合**（ $AA$ ）的概率为  $\frac{1}{4}$ ，且各位点相互独立地遗传。

他们生育了  $n$  个孩子。我们称一个孩子是**完美的**，当且仅当他在全部  $m$  个位点上都是显性纯合（即  $m$  个位点全部为  $AA$ ）。不同孩子的基因型相互独立。

请你回答：这  $n$  个孩子中，恰好有  $k$  个孩子是完美的，概率是多少？

由于答案是一个有理数，可以证明它可以表示为最简分数  $p/q$  且  $q \not\equiv 0 \pmod{998244353}$ ，你需要输出  $p \cdot q^{-1} \bmod 998244353$ 。

### 输入格式

第一行一个整数  $q$  ( $1 \leq q \leq 10^5$ )，表示询问组数。

接下来  $q$  行，每行三个整数  $m, n, k$  ( $1 \leq m \leq 10^9$ ,  $0 \leq k \leq n \leq 10^{18}$ )，含义如上。

保证所有询问的  $k$  之和不超过  $2 \times 10^6$ 。

### 输出格式

对每组询问输出一行一个整数，表示所求概率对 998244353 取模的结果。

### 样例

| 标准输入  | 标准输出      |
|-------|-----------|
| 3     | 748683265 |
| 1 1 1 | 577110017 |
| 1 4 1 | 977681153 |
| 2 4 2 |           |

## Problem K. LittleQiu 的历史作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 1 s  
内存限制 256 MB

LittleQiu 收到了他的历史作业，作业内容如下：

我国古代数学名著《孙子算经》中记载了一道著名的题目——「物不知数」：

今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问物几何？

用现代的语言来说，就是要找一个整数  $x$ ，使得它除以 3 余 2、除以 5 余 3、除以 7 余 2。  
后世将求解这类问题的方法称为「孙子定理」，民间也流传着「韩信点兵」的故事。

现在给你  $n$  个这样的条件，第  $i$  个条件为「 $x$  除以  $m_i$  余  $r_i$ 」，即

$$x \equiv r_i \pmod{m_i}.$$

请你求出满足所有条件的最小非负整数  $x$ 。如果不存在这样的  $x$ ，输出  $-1$ 。

### 输入格式

第一行一个整数  $n$  ( $1 \leq n \leq 10^5$ )，表示条件个数。

接下来  $n$  行，每行两个整数  $r_i, m_i$  ( $1 \leq m_i \leq 10^{18}$ ,  $0 \leq r_i < m_i$ )，表示一个同余条件。

保证所有模数  $m_i$  的最小公倍数不超过  $10^{18}$ 。

### 输出格式

输出一行一个整数，表示满足所有条件的最小非负整数  $x$ ；若无解，输出  $-1$ 。

### 样例

| 标准输入                   | 标准输出 |
|------------------------|------|
| 3<br>2 3<br>3 5<br>2 7 | 23   |
| 标准输入                   | 标准输出 |
| 2<br>0 2<br>1 2        | -1   |

### 提示

样例 1 即《孙子算经》原题：23 除以 3 余 2、除以 5 余 3、除以 7 余 2，且是满足条件的最小非负整数。

样例 2 要求  $x$  既是偶数又是奇数，显然不存在，故输出  $-1$ 。

## Problem L. LittleQiu 的地理作业

输入文件 标准输入  
输出文件 标准输出  
时间限制 3 s  
内存限制 256 MB

LittleQiu 收到了他的地理作业，作业内容如下：

小 C 是一名地理测绘员。他负责的一片矩形辖区被绘制成方格地图，图上已经标注了  $n$  个界桩（测量控制点）。上级要求他在辖区内划定一块**自然保护区**，希望保护区的面积尽可能大，且保护区内部不能圈入任何已有界桩（界桩落在保护区的边界或顶点上则允许）。为便于测量与立牌，保护区必须是一个矩形，且四条边分别与辖区的外围平行。小 C 很快发现，为了让矩形的四个角都恰好对准控制点，他不得不在地图上补设一些新的界桩，但他不想设立太多新界桩，每补设一根新界桩都要投入一定的开销。请你帮小 C 规划一个矩形保护区，他的收益是保护区面积减去补设新界桩的总开销，他的目标是让收益尽量大。

辖区是一个矩形，左下角坐标为  $(0,0)$ ，右上角坐标是  $(l,r)$ ，其中  $l,r$  均为整数。界桩的坐标均为整数，界桩可以位于辖区的边界上。小 C 划定的矩形保护区，其**四个顶点都必须有界桩**（原有的或新补设的均可）；保护区内部（不含边和顶点）不能有任何界桩。小 C 补设的新界桩必须在整数坐标，可以立在辖区的边界上。每根新界桩的开销为  $w$ 。保护区必须有正的面积。

### 输入格式

输入第一行包含四个正整数  $l,r,w,n$  ( $1 \leq l,r,w \leq 10^9, 1 \leq n \leq 1000$ )，含义如题目所描述。

接下来  $n$  行，每行包含两个整数  $x_i, y_i$ ，满足  $0 \leq x_i \leq l, 0 \leq y_i \leq r$ ，表示每个界桩的坐标。数据保证所有界桩的坐标互不相同。

### 输出格式

输出一个整数，为小 C 的最大收益。该收益可以为负值。

### 样例

| 标准输入                                              | 标准输出 |
|---------------------------------------------------|------|
| 5 5 3 6<br>0 0<br>0 2<br>1 4<br>2 3<br>3 2<br>4 1 | 3    |
| 标准输入                                              | 标准输出 |
| 5 5 10 1<br>2 2                                   | -21  |

## 提示

**样例 1:** 在  $(3, 0)$  处补设一根界桩，以  $(0, 0), (3, 0), (3, 2), (0, 2)$  为顶点划出面积为 6 的矩形保护区。四个顶点中仅  $(3, 0)$  需补设，开销 3，总收益  $6 - 3 = 3$ 。

**样例 2:** 以  $(2, 2), (5, 2), (2, 5), (5, 5)$  为顶点划出面积为 9 的矩形保护区，需补设其余 3 个角，开销 30，收益  $9 - 30 = -21$ 。