北京理工大学第七届程序设计新生赛

The 7-th BIT Campus Programming Contest for Junior Grade Group

现场赛

Onsite Round

题目列表

Problem List

A	自然语言处理
B	The Secret of Time
C	Lytchen loves JSON
D	元素周期表
E	龙语魔法
F	浴缸
G	伙伴系统
H	白学串
I	Integer Factorization
J	机房的圣诞礼物
K	X-Window System
L	Bonus Quiz
M	长安街的华灯

请勿在比赛开始前翻阅试题!

Do not open before the contest has started.

2018 年 12 月 30 日

Problem A. 自然语言处理

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	64 megabytes

想必刚刚学完线性代数的你，深知它在数学代数领域的重要性，下面就来练练手吧！

在自然语言处理领域，有一种算法叫做词嵌入向量（*Word Embedding*）法，算法的核心便是将语言抽象成代数中矩阵的形式进行运算。而其中一种很重要的方法便是词频统计法，该方法通过统计一段文本中每个单词出现的频数、建立文本段落的词频向量来确定文本的特征。其中词频向量是一个 $1 \times m$ 的矩阵，其中每个分量都对应着某个单词的出现次数。

下面将给出若干段由小写英文字母组成的文本内容，请你通过上文描述的词频统计法计算出每段文本对应的词频向量，再判断这些向量所构成的向量组是否满足线性相关性质.....

但是，善良的龙龙不忍心让你做这么繁琐又无聊的字符串处理工作。于是他偷偷帮你处理出了所有文本的词频向量，剩下的工作交给你了！

Input

第一行输入一个正整数 T $(1 \leq T \leq 100)$，表示数据组数。

接下来 T 组数据，每组数据将输入 $n+1$ 行，第一行输入两个正整数 n $(1 \leq n \leq 10)$ 和 m $(1 \leq m \leq 4)$，分别表示有 n 段文本，他们的词频向量维度都是 m。

接下来 n 行，每行输入 m 个非负整数，相邻两个整数由空格间隔开，其中第 i 行的输入表示龙龙帮你处理完的第 i 段文本的词频向量是 $A[i] = \{a_1, a_2, \cdots, a_m\}$ 的形式，保证 $0 \leq a_i \leq 100$。

Output

对于每组数据，请输出一行，如果这些词频向量组满足线性相关的性质，则请输出 `YES`，如果它们线性无关则请输出 `NO`，注意换行。

Example

standard input	standard output
1 2 2 1 1 0 1	NO

Explanation

容易发现，第一组词频向量和第二组词频向量线性无关。

Problem B. The Secret of Time

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

As the galaxy's *Supreme Overlord* annihilated metal gloves gathered all the infinite gems, the heroes of the Avengers all turned to ashes as his fingertips rang.

In order to save the hero of sacrifice, and to reverse the pattern of changing the world in time, you took the time wheel to the end of the universe. The door to the end of the universe is a password lock...

At this time, you heard the voice of an elder wearing black-rimmed glasses whisper: The secret of time, which is a magical number. The result of using the square operation in this magical number is the ultimate answer of the universe. The ultimate answer is a string of 16-digit Arabic number. For this natural number, when counting it from the right to the left, the first and thirteenth digits are 1, and the third digit is 9, and the fifth digit is 2, and the seventh digit is 6, and the ninth digit is 0, and the eleventh digit is 8, and the fifteenth digit is 7...

The world is in danger, and you have no time to left, please start to crack the secret of time lock!

Input

This problem has no input.

Output

Please output a password which can be the key to open the time gate password lock.

Example

standard input	standard output
(No input)	32768

Note

The secret of time may not be unique, any of them should be an answer for this lock, since it's no wonder that the universe is so beautiful and mystery!

Problem C. Lytchen loves JSON

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	128 megabytes

JSON 的全称为 JavaScript Object Notation，是一种用于快速数据交换的轻量级数据对象编码结构格式，由于其解析方便，且对人类友好，如今已被广泛用于各种应用服务数据交换场景。下面将着重阐述 JSON 对数据对象的编码结构。

JSON 使用键值对集合的形式对一个数据对象的各个属性进行编码。例如，考虑如下的用于存储一个学生信息的数据结构（伪代码）：

```
structure Student {
    name: String
    gender: String
    age: Integer
}
```

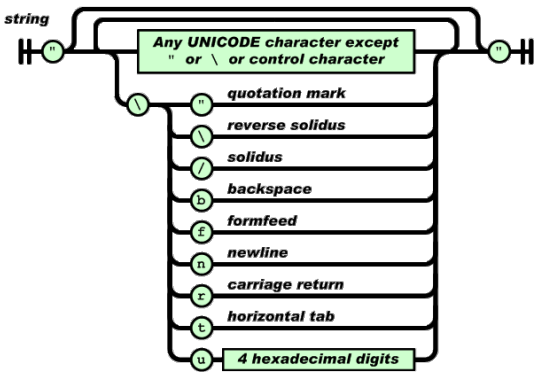
则一个实际的数据对象可能以如下的 JSON 格式进行编码：

```
{
    "name": "Lytning",
    "gender": "man",
    "age": 19
}
```

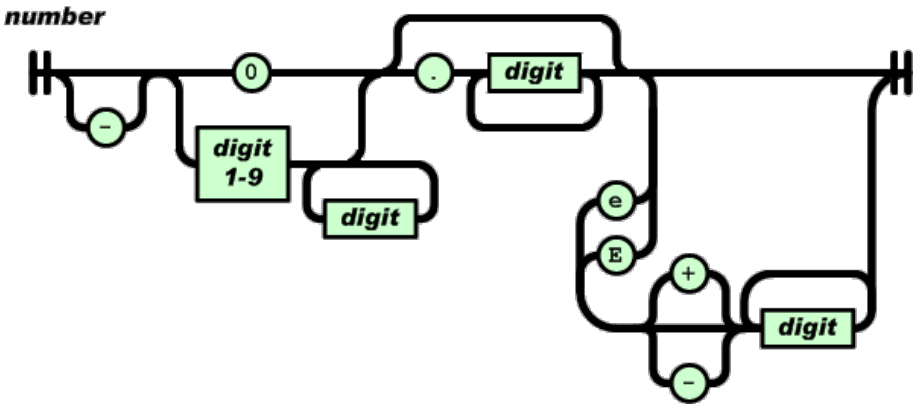
可以看到，JSON 使用键值对的方式对一个数据对象的各个属性进行编码。一个数据对象被一对花括号包围，花括号之中是一系列的 `"name":"value"` 形式的键值对，表示该数据对象的属性 `name` 具有值 `value`。

需要注意到的是，在键值对集合中，每一个“键”均需要以双引号进行包裹，但是其值不一定需要以双引号进行包裹。在键值对中，值的编码方式由值的类型决定。JSON 中的值可能有如下的类型：

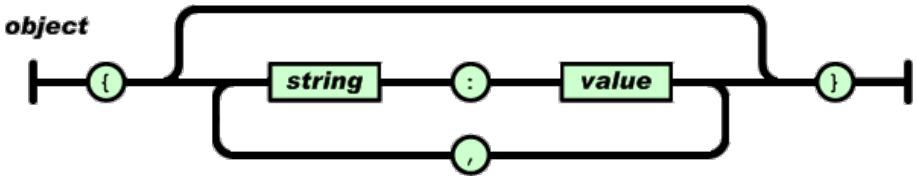
- **字符串类型**。此时值需要被一对双引号进行包裹，与上述的例子相同。在字符串中支持转义字符，其转义规则与 C 语言一致。例如，`\t` 表示制表符，`\n` 表示换行符，`\r` 表示回车等。字符串类型的编码逻辑如下图所示：



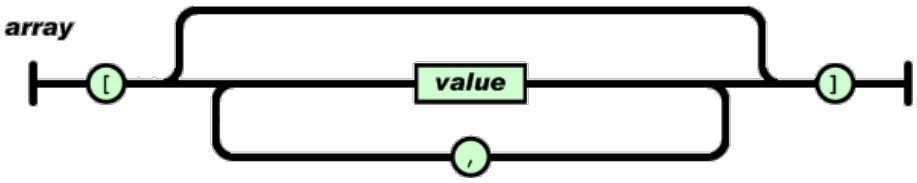
- **数字类型**。由于 JSON 的类型系统继承于 JavaScript 的类型系统，因此 JSON 中并不区分整数与浮点数。所有数字均被视为浮点数进行存储。数字类型的编码规则与 C 语言中数字的编码规则十分类似，但 JSON 不支持 C 语言中的字面八进制数字和十六进制数字的编码方式。数字类型的编码逻辑如下图所示：



- **布尔类型**。在布尔类型中，合法的值只有两个，即 `true` 和 `false`。
- **对象类型**。此时值的编码采用题目开头给出的键值对集合的方式进行编码，会产生嵌套的大括号。对象类型的编码逻辑如下图所示：



- **数组类型**。在 JSON 中，数组由一组中括号进行包裹，其中包含了一组以逗号分隔的值。JSON 的数组是不同质的。也就是说，JSON 数组中可能包含多种类型的值。数组类型的编码逻辑如下图所示：



- 另外，JSON 中包含一个特殊的“空”值，以 `null` 进行编码。

下面一个示例为大家展示每种类型的编码结构。该示例将基于下面的数据对象结构。

```
structure Date {
    year: Number
    month: Number
    day: Number
}
structure Student {
    name: String
    gender: Boolean
```

```
    height: Number
    birthday: structure Date
    grades: Array,
    laboratory: String
}
```

一个 JSON 编码实例如下。该 JSON 的最外层编码一个 Student 对象。

```
{
    "name": "Lchen",
    "gender": false,
    "height": 1.60e+2,
    "birthday": {
        "year": 2000,
        "month": 1,
        "day": 1
    },
    "grades": [ 90, 80, 88, [ 55, 80 ] ],
    "laboratory": null
}
```

好了，现在你已经掌握了使用 JSON 对数据对象进行编码的基本方法。下面让我们使用这些理论解决一个实际的问题吧。

在这一个任务中，你需要编写一个程序，这个程序以一个合法的 JSON 文档作为输入，然后响应一系列查询。每个查询均会要求查询在这个 JSON 文档所包含的对象图上的一个值。

有关具体的查询细节，请参见输入输出格式以及样例。

Input

输入文件从头开始的若干行包含一个合法的 JSON 文档。这个 JSON 文档的最顶层元素一定是一个对象，即这个 JSON 文档在最外层是被一对大括号包裹的键值对集合。

接下来直到文件末尾前的每一行均包含一个查询字符串。每个查询以一系列以英文句点“.”分隔的属性名表示，每个属性名的末尾可能包含一个或多个分别以中括号包裹的非负整数，表示一次下标操作。例如，查询字符串 A.B.C[0].D[0][1].E 表示沿着对象图中的以下路径进行值查询：最外层对象的 A 属性的值 → B 属性的值 → C 属性的值 → 在下标为 0 位置处的值 → D 属性 → 在下标为 0 位置处的值 → 在下标为 1 处的值 → E 属性。

保证 JSON 文档不超过 100 行，每行不超过 100 个字符。保证 JSON 文档中每一个键值对的键仅由英文字母组成。保证输入的 JSON 文档中的每个字符串值在字面上仅包含 ASCII 字符。保证 JSON 文档中在相同键值对集合中的每个键互不相同。保证 JSON 文档语法完全合法，但不保证 JSON 文档的格式。

保证查询的数量不超过 100，每个查询字符串的长度均不超过 100，每个查询字符串中不包含任何空白字符。当查询字符串在字符串类型的值上进行下标运算时，保证通过下标获取到的字符是非空白 ASCII 字符。

Output

对于每个查询，输出一行表示查询结果。

若要查询的目标属性在对象图中不存在，输出一行 `Error: no such attribute`；若查询字符串中带有下标运算而目标属性的类型不是字符串或者数组类型，输出一行 `Error: invalid type`；若查询字符串中带有下标运算但下标溢出，输出一行 `Error: index overflow`。

对于每一种 JSON 类型，其输出格式如下：

- 布尔类型直接输出 `true` 或者 `false` 即可。
- 数字类型以定点小数的形式进行输出，保留到小数点后两位，不足的部分以 0 补齐。
- 字符串类型的数据直接输出即可。在输出时，仅需要对 `\t`、`\\`、`\/` 与 `\"` 转义序列进行转义，其它转义序列按原样输出即可。对字符串类型的数据进行下标运算之后的结果以单个字符的形式进行输出。
- 特殊的“空”值直接输出 `null` 即可。
- 数组类型的值首先输出一个左中括号“[”，然后输出一个空格，然后输出每一个数组元素。数组元素之间使用一个逗号以及一个空格进行分隔。最后，输出一个空格以及右中括号“]”。
- 对象类型的值首先输出一个左花括号“{”，然后输出一个空格，然后输出所有的属性键值对。在键值对中的冒号之后应输出一个空格。同一个对象的所有属性键值对之间以逗号以及一个空格分隔，同一个对象的所有属性键值对应该以键的字典序顺序排序后再进行输出。最后，输出一个空格以及右花括号“}”。

注意，所有的值均在一行上进行输出。

Example

standard input	standard output

{
"name": "Lchen",
"gender": false,
"height": 1.60e+2,
"birthday": {
"year": 2000,
"month": 1,
"day": 1,
"aggregate": [2000, 1, 1]
},
"grades": [90, 80, 88, 100, [55, 80]],
"laboratory": null,
"description": "\t\\\\The quick brown \"fox\" jumps over the lazy dog\u25A0"
}
name
name[0]
name.gender
gender
gender[1]
height
birthday
birthday.year
grades
grades[2]
grades[5]
laboratory
grades[4]
grades[4][1]
description
grades[0].name
teammates
teammates[0]
	Lchen
	L
	Error: no such attribute
	false
	Error: invalid type
	160.00
	{ "aggregate": [2000.00, 1.00 1.00], "day": 1.00, "month": 1.00, "year": 2000.00 }
	2000.00
	[90.00, 80.00, 88.00, 100.00, [55.00, 80.00]]
	88.00
	Error: index overflow
	null
	[55.00, 80.00]
	80.00
	\\The quick brown "fox" jumps over the lazy dog\u25A0
	Error: invalid type
	[“Lytning”]
	L

Problem D. 元素周期表

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

龙神除了打 ACM 之外，最擅长的就是化学了，擅长化学当然要能熟背元素周期表辣！看，龙神分分钟就又背了一遍元素周期表：氢氦锂铍硼，碳氮氧氟氖，钠镁铝硅磷、硫氯氩钾钙……

龙神背完元素周期表，又开始说起了关于元素周期表的段子。

龙神：你们知道居里夫人一生中发现了几种元素吗？

我们：母鸡啊。

龙神：居里夫人一共发现了 2 种元素，你们知道这 2 种元素都是什么吗？

我们尴尬地摇了摇头。

龙神：一个是“镭”，镭元素的英文名是 Radium，是根据辐射（radiation）这个词命名的，因为镭元素有很强的放射性；另一个是“钋”，英文名是 Polonium，是根据居里夫人的国籍波兰（Poland）命名的。

我们：龙神太强辣！

龙神：其实还有一个元素，是以居里夫人（Curie）命名的，就是“锔”元素，Curium。这个元素是加州伯克利分校发现的，加州伯克利分校还发现了另外 3 种元素，分别是锫（Berkelium），锎（Californium），镅（Americium），这三种元素分别以伯克利（Berkeley）、加州（California）和美国（America）命名。说到这几种元素的中文译名，就更有意思了。Americium 是以没过命名的金属元素，那为什么不叫“镁”呢？显然，在给镁元素命名的时候，万万没想到会有以美国命名的元素，但是又不能把之前起的名字改了，所以只能将就一下，让 Americium 叫“镅”了。Curium 是以居里夫人命名的金属元素，为什么不用“锯”来命名呢？仔细一看，拿“锯”来命名元素还得了，这个简直比镅命名的时候还尴尬。

龙神说完段子，终于想起来一件要紧事，那就是出新生赛的题。龙神想了想，既然你们的化学学得这么差，那我就出个签到题吧。

龙神虽然化学很强，但是他最讨厌查表算分子量了，于是龙神抛给了大家一坨分子式，想要大家来计算这些分子式的相对分子质量。

Input

第一行输入一个正整数 T ($1 \leq T \leq 200$)，表示数据组数。接下来 T 组数据，每组数据输入一行，包含一个分子式，保证满足以下条件：

- 分子式只包含元素和元素个数（不会出现形如 Al2(SO4)3 这样包含括号的形式或 CuSO4 • 5H2O 这样带有前导数字的形式）；
- 每个元素后的数字不超过 1000；
- 分子式总长度不超过 1000。

Output

对于每组数据，请输出一个浮点数，表示该分子式的分子量。输出结果要求精确到 10^{-3}，若精确答案为 a，你的输出结果为 b，只要满足 $|a-b|<10^{-3}$，评测法官 Jury 就认为你的输出是正确的，注意换行。

Example

standard input	standard output
2 H2O CO2	18.016 44.01

Note

周期\族

	IA												IIIA	IVA	VA	VIA	VIIA	VIIIA(0)
1	1 H 氢 1.008	IIA																2 He 氦 4.003
2	3 Li 锂 6.941	4 Be 铍 9.012											5 B 硼 10.81	6 C 碳 12.01	7 N 氮 14.01	8 O 氧 16.00	9 F 氟 19.00	10 Ne 氖 20.18
3	11 Na 钠 22.99	12 Mg 镁 24.31	IIIB	IVB	VB	VIB	VIIB		VIIIB		IB	IIB	13 Al 铝 26.98	14 Si 硅 28.09	15 P 磷 30.97	16 S 硫 32.07	17 Cl 氯 35.45	18 Ar 氩 39.95
4	19 K 钾 39.10	20 Ca 钙 40.08	21 Sc 钪 44.96	22 Ti 钛 47.88	23 V 钒 50.94	24 Cr 铬 52.00	25 Mn 锰 54.94	26 Fe 铁 55.85	27 Co 钴 58.93	28 Ni 镍 58.69	29 Cu 铜 63.55	30 Zn 锌 65.39	31 Ga 镓 69.72	32 Ge 锗 72.59	33 As 砷 74.92	34 Se 硒 78.96	35 Br 溴 79.90	36 Kr 氪 83.80
5	37 Rb 铷 85.47	38 Sr 锶 87.62	39 Y 钇 88.91	40 Zr 锆 91.22	41 Nb 铌 92.91	42 Mo 钼 95.94	43 Tc 锝 97.91	44 Ru 钌 101.1	45 Rh 铑 102.9	46 Pd 钯 106.4	47 Ag 银 107.9	48 Cd 镉 112.4	49 In 铟 114.8	50 Sn 锡 118.7	51 Sb 锑 121.8	52 Te 碲 127.6	53 I 碘 126.9	54 Xe 氙 131.3
6	55 Cs 铯 132.9	56 Ba 钡 137.3	57-71 镧系元素	72 Hf 铪 178.5	73 Ta 钽 180.9	74 W 钨 183.9	75 Re 铼 186.2	76 Os 锇 190.2	77 Ir 铱 192.2	78 Pt 铂 195.1	79 Au 金 197.0	80 Hg 汞 200.6	81 Tl 铊 204.4	82 Pb 铅 207.2	83 Bi 铋 209.0	84 Po 钋 209.0	85 At 砹 210.0	86 Rn 氡 222.0
7	87 Fr 钫 223.0	88 Ra 镭 226.0	89-103 锕系元素	104 Rf 𨧀 265.1	105 Db 𨨏 268.1	106 Sg 𨭆 271.1	107 Bh 𨨨 270.1	108 Hs 𨭉 277.2	109 Mt 鿏 276.2	110 Ds 𫟼 281.2	111 Rg 𬬭 280.2	112 Cn 鿔 285.2	113 Nh 鿭 284.2	114 Fl 𫓧 289.2	115 Mc 镆 288.2	116 Lv 鉝 293.2	117 Ts 294.2	118 Og 294.2

镧系元素	57 La 镧 138.9	58 Ce 铈 140.1	59 Pr 镨 140.9	60 Nd 钕 144.2	61 Pm 钷 144.9	62 Sm 钐 150.4	63 Eu 铕 152.0	64 Gd 钆 157.3	65 Tb 铽 158.9	66 Dy 镝 162.5	67 Ho 钬 164.9	68 Er 铒 167.3	69 Tm 铥 168.9	70 Yb 镱 173.0	71 Lu 镥 175.0
锕系元素	89 Ac 锕 227.0	90 Th 钍 232.0	91 Pa 镤 231.0	92 U 铀 238.0	93 Np 镎 237.1	94 Pu 钚 244.1	95 Am 镅 243.1	96 Cm 锔 247.1	97 Bk 锫 247.1	98 Cf 锎 252.1	99 Es 锿 252.1	100 Fm 镄 257.1	101 Md 钔 258.1	102 No 锘 259.1	103 Lr 铹 262.1

Problem E. 龙语魔法

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	64 megabytes

龙老师新学到一个龙语魔法，他非常高兴。

对于一个有序数字序列 $a_1, a_2, \cdots, a_n$ 共 n 个数字，龙老师的新魔法可以瞬间得到数字序列的某一**连续段**的和，即序列的某个子串和。他只需要念咒语 `orz! l r`，其中 l 和 r 是要得到的连续段的左右边界，就可以得到 $\sum_{i=l}^{r} a_i$，但是应当保证 $1 \le l \le r \le n$。

龙老师觉得这个魔法很好，并且玩得很开心。但是龙老师发现，这个魔法能产生的结果值有很多，其中有一些数字可能在结果中出现，有些数字却可能不会在结果中出现。现在龙老师想知道，在所有能产生的结果中，第 k 小的数字是多少？聪明的你快帮帮龙老师吧！**如果同一个数字在结果中出现多次，那么它也会被统计多次。**

Input

输入共两行，第一行输入两个正整数 n 和 k $(1 \le n \le 10^5, 1 \le k \le \frac{n(n+1)}{2})$，表示龙老师的序列长度为 n，他想知道第 k 小的结果。

接下来一行输入 n 个正整数 $a_1, a_2, \cdots, a_n$ $(1 \le a_i \le 10^9)$，相邻两个正整数由空格间隔开，依次表示序列中的数字内容。

Output

请输出第 k 小的结果值，注意换行。

Example

standard input	standard output
4 6 2 3 1 4	5

Explanation

对于样例而言，魔法能产生的所有结果是 $\{1, 2, 3, 4, 4, 5, 5, 6, 8, 10\}$ 。其中第 6 小的数字是 5。

Problem F. 浴缸

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	64 megabytes

众所周知龙龙虽然很可爱但其实是俱乐部最强的人。这不他今年去打 EC-Final 辣。处于一些特殊原因，龙龙订了一间主题房，里面有一个大大的浴缸。虽然龙龙很想使用这个浴缸但是傲娇的龙龙打算掩盖自己的想法。他问大佬，假设已知这个浴缸的尺寸，如果往这个浴缸里注入 n 升（单位：L）水，那么从浴缸顶部到水面的距离有多长。

对于浴缸尺寸的描述，我们将浴缸的俯视图视为一个矩形，将浴缸从俯视图上按单位长度分为一个个方格，对每个方格有一个数字 $h_{i,j}$ 分别描述从浴缸顶部到当前位置底部的距离。

Input

第一行输入两个正整数 n, m, v 由空格间隔开，保证 $1 \le n, m \le 1000, 1 \le v \le 10^9$，表示这个浴缸的长与宽，以及注入的水的体积。

接下来 n 行每行输入 m 个数字，其中第 i 行第 j 个数字是 $h_{i,j}$ $(1 \le h_{i,j} \le 100)$，表示浴缸在这一块方格中从顶部到底部的距离。

Output

请输出一个非负整数，表示浴缸从顶部到水面的距离。傲娇的龙龙其实早就在心里想了好多泡澡的事情辣，所以可以保证输出一定是一个整数，并且一定不会有水溢出，他早就考虑好辣。

Example

standard input	standard output
4 3 6 2 2 1 2 2 1 2 2 1 1 1 1	1
3 3 15 1 2 3 4 5 6 7 8 9	4

Note

所有的长度单位都是分米，分米和升的单位转换关系是：1 立方分米（单位：dm^3）= 1 升（单位：L）。

而且保证浴缸矩阵是非严格单调的，即保证不会出现水填满了其中的一部分但有一部分比水平面低的地方没有被水填充的情况。

Explanation

对于样例 1 的浴缸图如下：

Problem G. 伙伴系统

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

龙神最近复习操作系统走火入魔了，为了取得好成绩，他决定放出一道签到题积攒人品。他将 Linux 内存管理的 Buddy System 简化为如下模型：

- 假设内存都以 KB 为最小单位分配，因此以下描述省略单位。
- 系统只记录大小恰为 $2^0, 2^1, \cdots, 2^{10}$ 的 11 种空闲内存块个数。
- 当获得大小为 n $(1 \le n < 2^{11})$ 的空内存块时，系统将其分解为上述 11 种大小的内存块，且分解出的内存块中每种大小至多出现一次。可以证明给定条件下这种分解是唯一的。
  - 例如，一个大小为 9 的空内存块被分解为一个大小为 $2^3 = 8$ 的空内存块和一个大小为 $2^0 = 1$ 的空内存块。
- 当请求分配大小为 m $(1 \le m \le 2^{10})$ 的内存块时，系统将挑选出尺寸大于等于 m 的最小的内存块（设其大小为 m_0）并将其分为两部分：一部分大小为 m 用于分配，另一部分大小为 $(m_0 - m)$ 则当作新获得的空内存块按上一条的规则重新记入系统。
  - 例如，请求分配一个大小为 13 的内存块时，系统将检测是否存在大小为 $2^4 = 16$ 的空内存块，若存在，则分为大小分别为 13 和 3 的两部分，前者用于满足请求，后者则分解为大小分别为 $2^0 = 1$ 和 $2^1 = 2$ 的两个空闲块放回空内存表。若不存在大小为 $2^4 = 16$ 的空内存块，则依次尝试大小为 $2^5, 2^6, \cdots, 2^{10}$ 的空内存块，都不存在则申请失败。
- 在本题中，多个小的空闲内存块不能合并（见样例 2）。这是因为其在内存中的位置不一定连续。

现在假设系统中当前没有空内存块。龙神要求你模拟 Buddy System 的一个长度为 k 的操作序列，共有两种操作，并要求你在每次操作后输出一行 11 个整数，由空格间隔开，代表 11 种空闲内存块的个数：

- `free n`，代表获得一块大小为 n $(1 \le n < 2^{11})$ 的内存。
- `allocate m`，代表请求分配一块大小为 m $(1 \le m \le 2^{10})$ 的内存，若分配失败，则输出一行 `ERROR!`，并忽略本次操作。

若操作成功，输出一行 11 个整数代表完成操作后 11 种空闲内存块的个数；若操作失败（即 allocate 分配失败），则输出一行 `ERROR!`。

Input

输入第一行一个正整数 k $(1 \le k \le 10^5)$，代表操作序列长度。

接下来 k 行每行将描述一个操作，格式见题目描述。

Output

每次操作处理完成后，若操作成功则请输出一行 11 个非负整数，代表 11 种空闲内存块的个数（行末无空格），若操作失败输出一行 `ERROR!`。

Example

standard input	standard output
3 allocate 1 free 1024 allocate 1	ERROR! 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0
5 free 1 free 1 free 1 free 2 free 2	1 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 3 1 0 0 0 0 0 0 0 0 0 3 2 0 0 0 0 0 0 0 0 0

Note

本题的输入和输出数据较多，如果你使用的编程环境是 `C` 语言或者是 `C++`，那么强烈建议你使用 `scanf` 函数和 `printf` 函数进行输入和输出操作，而不是使用 `std::cin` 和 `std::cout` 进行输入和输出操作，否则评测法官 Jury 可能会返回给你一个 `TIMELIMIT` 作为评测结果。

Problem H. 白学串

Input file:	standard input
Output file:	standard output
Time limit:	1 seconds
Memory limit:	64 megabytes

龙老师特别擅长出各种奇奇怪怪而且很难的题目。今天龙老师又想了一道好玩的题目。

白学家龙老师有一个数字序列 $a_1, a_2, \cdots, a_n$，定义这个序列的子串是序列中的连续一段，也就是 $a_x, a_{x+1}, \cdots, a_{y-1}, a_y$ $(1 \le x \le y \le n)$。定义一个子串是白学的，当且仅当这个子串中存在 3 个元素 a_i, a_j, a_k $(x \le i, j, k \le y)$ 能够以它们为边长组成一个合法的三角形。现在龙老师想知道，对于他询问的每个子串，那些是白学串。

Input

第一行输入一个正整数 T $(1 \le T \le 100)$，表示数据组数。输入保证 $\sum n \le 5 \times 10^5, \sum m \le 10^6$。

接下来 T 组数据，每组数据第一行输入两个正整数 n, m $(1 \le n \le 10^5, 1 \le m \le 2 \times 10^5)$ 由空格分隔开，分别表示数字序列长度和询问数。

第二行输入 n 个正整数，由空格间隔开，其中第 i 个正整数是 a_i $(1 \le a_i \le 10^8)$，描述这个序列。

接下来 m 行，每行输入两个正整数 l 和 r $(1 \le l \le r \le n)$ 由空格分隔开，表示龙老师想问你：从第 l 个到第 r 个数组成的子串 $a[l \cdots r]$ 是不是白学的。

Output

对于每组数据，请输出 m 行，表示 m 个询问的答案。如果这个子串是白学的，请输出 `yes`，否则请输出 `no`。

Example

standard input	standard output
2 5 3 1 1 2 3 4 1 3 2 4 2 5 5 2 2 3 4 1 2 1 4 3 5	no no yes yes no

Explanation

对于第一组数据的 3 个询问，$\{1, 1, 2\}$ 和 $\{1, 2, 3\}$ 都不存在 3 个数字能构成三角形，$\{1, 2, 3, 4\}$ 可以选出 $2, 3, 4$ 构成三角形。

Problem I. Integer Factorization

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

在数学中，整数分解（Integer Factorization）又称素因数分解（Prime Factorization），是将一个正整数写成几个约数的乘积。根据算术基本定理，这样的分解结果应该是独一无二的。这个问题在代数学、密码学、计算复杂性理论和量子计算机等领域中有重要意义。非对称性加密 RSA 的安全性就是仰赖于整数分解的难度，生成 RSA 密钥的第一步是寻找两个大质数 p 和 q，计算 $n = p * q$，如果 n 被成功分解那么就意味着这个 RSA 被破解了。

每个学过 C 语言的同学都写过使用试除法进行因式分解的程序，但是当需要分解的因数很大的时候，这种方法就显得捉鸡了。现在有很多优秀的算法可以进行因数分解，如果 $p+1$ 或者 $p-1$ 光滑的时候可以使用 *Pollard's* $p-1$ *算法*和 *Williams'* $p+1$ *算法*，更加通用的方法是 *Lenstra 椭圆曲线分解法*。

但是龙神觉得这都没东西，给出一个 n，然后你敲个板子就能跑出来 p 和 q，这样的模板题放在新生赛实在是没什么意思，而且对没有带模板的选手实在不公平。因此龙神在 n 上多安排了一些特效：给定两个大整数 a 和 b（保证大整数在 C 语言的 `long long int` 范围内），p 和 q 都是质数，而且满足：

$$\begin{cases} a = (p * q) \oplus (p + q) \\ b = (p * q) \oplus (p - q) \end{cases}$$

其中 $\oplus$ 表示对于整数使用按位异或操作，例如在 C++ 和 Java 语言中对应的是 `xor` 操作符，对于每一位的二进制遵循如下操作：0 xor 0 = 0, 0 xor 1 = 1, 1 xor 0 = 1, 1 xor 1 = 0。

龙神相信聪明的你一定非常熟悉二进制的异或操作那一套理论，现在请你算一算原来的 p 和 q 是多少呢？

Input

第一行输入一个正整数 T $(1 \le T \le 3000)$，表示数据组数。

接下来 T 组数据，每组数据输入只有一行，即输入两个整数 a, b $(-2^{63} < a, b < 2^{63})$，由空格间隔开。

Output

对于每组数据，请输出两个整数，表示你计算的答案，先输出 p，再输出 q，两个整数由空格间隔开，注意换行，题目保证 p 和 q 的答案有唯一性。

Example

standard input	standard output
1 1279 1201	39 31

Explanation

对于样例，$39 * 31 \oplus (39 + 31) = 1279, 39 * 31 \oplus (39 - 31) = 1201$。

Problem J. 机房的圣诞礼物

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	64 megabytes

龙老师很喜欢从生活中寻找出题的灵感。这不，刚刚过去的圣诞节也不例外。

在平安夜的晚上，龙老师准备了一颗精美装饰的圣诞树藏在机房的某处角落，上面挂满了 n 件礼物，礼物们都被龙老师用精美的礼袋包装着，编号从 1 到 n，其中第 i 件礼物袋的编号为 i。

当某个小朋友发现了这颗圣诞树，TA 就可以选择一些心仪编号的礼袋作为自己的圣诞礼物。但是龙老师有一个要求，如果某位小朋友选择了编号为 x 的礼袋，那么 TA 将无法再选择编号为 $2x$ 的礼袋作为礼物了 (??? 黑人问号.jpg)，换句话说，第 i 个发现圣诞树的小朋友获得的礼物集合 S_i 满足：$\forall x \in S_i,\ 2x \notin S_i$。

定义小朋友们今晚平安夜的愉悦值 F 是 TA 获得的礼物集合中所有礼袋编号的总和，也就是有：第 i 个发现圣诞树的小朋友的愉悦值 $F_i = \sum_{x \in S_i} x$。那么问题来了，第一位发现圣诞树的小朋友可能的最大愉悦值 $\max\{F_1\}$ 是多少呢？

Input

输入共一行，只有一个数字 n ($1 \le n \le 10^5$)，表示龙老师准备了 n 件由精美礼袋包装好的圣诞礼物并挂在圣诞树上。

Output

请输出一个非负整数，表示最早发现圣诞树的小朋友可能的愉悦值 F_1 的最大值。

Example

standard input	standard output
5	13

Explanation

龙老师准备了 5 件礼物，礼袋上编号依次为 1, 2, 3, 4, 5，当第一位发现圣诞树的小朋友选择编号为 1, 3, 4, 5 的礼袋时，TA 的愉悦值 $F_1 = 1 + 3 + 4 + 5 = 13$，是所有可能情况中的最大值。

Problem K. X-Window System

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	128 megabytes

现如今，所有的面向终端用户的操作系统几乎都支持 GUI 用户交互方式。我们最为熟悉的 GUI 交互方式通常都是以“窗口”的方式对内容进行组织的，即应用程序将与用户的交互部分包含在一个个的屏幕窗口之中，用户可以通过窗口分辨正在运行的不同的应用程序并通过键盘、鼠标等输入设备分别与之进行交互。

为了简化问题，我们将窗口系统进行如下的抽象和约定：

- **窗口（Window）**是**屏幕视口（Viewport）**的一个子矩形区域。屏幕视口即屏幕设备上能够呈现画面的矩形部分。窗口所在的子矩形区域的边缘与屏幕视口矩形边缘平行；
- **屏幕坐标系（Screen Coordinate）**用于对屏幕上的窗口以及鼠标进行定位。屏幕坐标系的原点位于屏幕左上角，其 x 轴正方向竖直向下，其 y 轴正方向水平向右；
- 所有的窗口在被渲染到屏幕上时仅保留其在屏幕视口内的部分。窗口在屏幕视口外的部分在渲染时不予考虑；
- 所有的窗口在渲染时均具有**唯一**的 z-index 值用于控制窗口的层叠渲染。z-index 值小的窗口会在渲染时遮挡住 z-index 值大的窗口。每个窗口被其它窗口遮挡的部分不予渲染。所有窗口的 z-index 构成集合 $[0, n-1] \cap \mathbb{Z}$，其中 n 表示系统中窗口的总数，而 $\mathbb{Z}$ 表示整数集合；
- 在任意时刻，系统中均存在唯一的一个**活动窗口（Active Window）**。当一个窗口由非活动窗口变为活动窗口时，其 z-index 被系统更改为 0，其余窗口的 z-index 依次递增 1，旧的活动窗口变为非活动窗口。因此当一个窗口变为活动窗口时，其将被提到所有窗口的最前端。在这个过程中，系统需要重绘一部分屏幕面积以展示新的活动窗口在之前被遮挡的部分。用户通过鼠标点击某一个窗口的渲染部分以将这个窗口激活。窗口的可视边界也视为窗口的渲染部分。

现在，给定系统中的所有窗口的位置信息，以及一个用户的鼠标点击序列。请编写一个程序，计算当用户每次单击鼠标时，窗口系统需要至少重新渲染多大的屏幕面积。

Input

输入共 $n+q+3$ 行，第一行输入两个由空格分隔的正整数 W 与 H $(0 < W, H \leq 2000)$，分别表示屏幕视口的宽度与高度。

第二行输入一个正整数 n $(1 \leq n \leq 10)$，表示系统中打开的窗口总数。

接下来 n 行，每行输入四个以空格为分隔的整数 x_i，y_i，w_i 与 h_i $(-H \leq x_i \leq H, -W \leq y_i \leq W, 0 < w_i \leq W, 0 < h_i \leq H)$，分别表示第 i 个窗口的左上角在屏幕坐标系下的位置以及窗口的宽度与高度。第 i 个窗口的 z-index 为 $i-1$。

接下来一行输入一个正整数 q $(1 \leq q \leq 10)$，表示用户点击鼠标的次数。

接下来 q 行，每行输入两个非负整数 u_i 与 v_i $(0 \leq u_i \leq W, 0 \leq v_i \leq H)$，分别表示用户第 i 次单击鼠标时鼠标所处的屏幕坐标位置。

Output

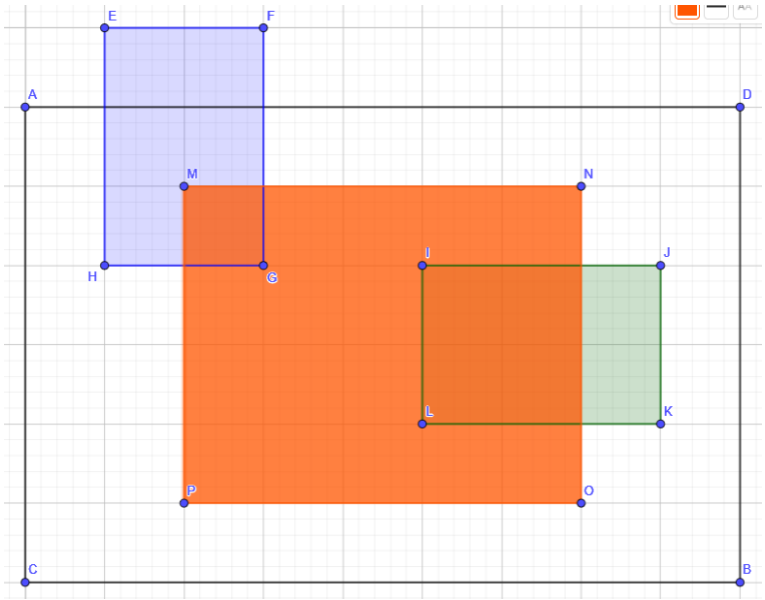
输出应包含 q 行，每行请输出一个非负整数，表示当用户单击第 i 次鼠标过后系统至少应该重绘多大的屏幕面积。

Example

standard input	standard output
9 6 3 1 2 5 4 2 5 3 2 -1 1 2 3 5 2 1 3 1 1 2 3 8 3 3	1 0 0 4 5

Explanation

在初始时，所有窗口以及屏幕视口的相对位置如下图所示：



在此图中，矩形 ACBD 表示屏幕视口。系统中包含了三个窗口，按照 z-index 递增序分别由矩形 MPON（窗口一）、矩形 ILKJ（窗口二）以及矩形 EHGF（窗口三）表示。矩形 EHGF 所表示的窗口（窗口三）有一部分超出了屏幕视口边缘（面积为 2），那一部分的窗口面积不会被渲染。

当用户执行了第一次鼠标点击，其点击位置为（2, 1）之后，由于屏幕坐标（2, 1）位于窗口三的渲染范围内，因此此时系统将活动窗口切换至窗口三。这个动作将导致窗口三的 z-index 值变为 0，窗口一与窗口二的 z-index 值分别变为 1 和 2。此时系统需要重绘窗口三之前被窗口一覆盖的屏幕部分，其面积为 1。因此第一个输出为 1。

当用户执行了第二次鼠标点击，其点击位置为（3, 1）之后，由于屏幕坐标（3, 1）不位于任何一个窗口的渲染范围内，因此此时系统将不会切换活动窗口。由于不会有任何活动窗口切换动作发生，因此系统需要重绘的屏幕面积为 0。

当用户执行了第三次鼠标点击，其点击位置为（1, 2）之后，由于此时屏幕坐标（1, 2）位于窗口三的渲染范围内（注意此时窗口三已经覆盖了窗口一，因此该屏幕坐标对应的窗口为窗口三而非窗口一），因此系统将会尝试将活动窗口切换至窗口三。由于窗口三已经是活动窗口，因此不会发生任何动作，屏幕重绘面积为 0。

当用户执行了第四次鼠标点击后，活动窗口切换到窗口二，重绘面积为 4。

当用户执行了第五次鼠标点击后，活动窗口切换到窗口一，重绘面积为 5。

Problem L. Bonus Quiz

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

As a league of *Fatty Newbie*, miamiao always dreams of making great fortune overnight.

Today, she comes to one lottery tickets shop hosted by Lytning and decides to choose several tickets which has continuous lottery numbers randomly. In other words, if the upper bound of this lottery round is n, she will choose two integers l and r ($1 \le l \le r \le n$) at random, and buy all the lottery tickets with the lottery numbers ranging from l to r.

However, you know that only a few tickets' number have bonus numbers, and these tickets are called *lucky tickets*. Lytning knows which lottery tickets have bonus numbers, please help him calculate the expectation value that miamiao could win exactly k lucky tickets.

It can be shown that the answer can be represented as $\frac{P}{Q}$, where P and Q are coprime integer numbers, and $Q \not\equiv 0 \ (mod\ 998244353)$. Print the value $P \cdot Q^{-1}$ modulo 998244353, where Q^{-1} denotes the modular inverse of Q modulo 998244353. It is guaranteed that this inverse exists and is unique.

Input

The first line contains three numbers n ($1 \le n \le 10^6$), m ($1 \le m \le n$) and k ($0 \le k \le m$) —— the upper bound of the number for this lottery round is n and the i-th ticket's lottery number is i, the number of lucky tickets is m and the number of lucky tickets miamiao hope to win after the lottery drawing is k.

The second line contains m numbers, the i-th number is b_i —— the lottery ticket with integer b_i is one lucky ticket.

Output

Print the expected calue in the format described in the problem statement.

Example

standard input	standard output
3 1 1 2	665496236

Explanation

In the example, miamiao has 6 ways to choose her lottery buying: $[1,1]$, $[1,2]$, $[1,3]$, $[2,2]$, $[2,3]$, $[3,3]$, the bonus number is 2.

Only the interval containing the bonus number 2 could satisfy her request —— $[1,2], [1,3], [2,2], [2,3]$. The exact expectation is $\frac{2}{3}$, so the answer is $2 \cdot 3^{-1} \ (mod\ 998244353) = 665496236$.

Problem M. 长安街的华灯

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	64 megabytes

来北京上学的你，一定会发现，笔直的长安街上，等距离排布着长长的华灯。当夜幕降临的时候，华灯初上，五彩的夜景点缀着美丽的长安大道，与天上的星星交相辉映。

在这个问题中，我们把华灯的彩色灯光能照亮范围看成是以灯杆为中心，半径为 R 的圆形区域，而且相邻两个灯杆的间隔为 L，请问当夜幕降临，华灯初上的长安街上，有哪些区域被照亮呢？

Input

第一行输入一个正整数 T $(1 \le T \le 10^5)$，表示数据组数。

接下来 T 组数据，每组数据第一行输入三个非负整数 N, R, L $(0 \le N, R, L \le 10^9)$，由空格间隔开，分别表示长安街上华灯的数量，每盏华灯能照亮的圆形区域半径，和相邻两盏华灯的间隔。

Output

对于每组数据，请输出一个浮点数，表示这些华灯能照亮的区域面积，注意换行。

输出结果要求精确到 10^{-6}，若精确答案为 a，你的输出结果为 b，只要满足 $\texttt{min}\{|\frac{a-b}{a}|, |a-b|\} \le 10^{-6}$，评测法官 Jury 就认为你的输出是正确的。

Example

standard input	standard output
1 4 1 1	8.881262

Explanation

对于样例，它的平面图如下，容易计算出所有被照亮的圆形面积是 4π，重叠的照亮区域面积是 $2\pi - \frac{3\sqrt{3}}{2}$，因此去除重叠的照亮区域以后答案是 $2\pi + \frac{3\sqrt{3}}{2} \approx 8.88$，也这些被照亮的圆形区域的面积并。