

2020 辽宁省 ACM 省赛 **大连大学** 名额赛
(正式赛)



主办单位：大连大学程序设计工作室

承办单位：大连大学 ACM 校队

2020 年 10 月 3 日

Problem A. Find Your Miku!!!

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



经过了各种维度的 Miku 寻找后，Miku 还是没有被找到。不过好消息是，这次我们又锁定了 Miku 的大概方位。

你的任务很简单，只需要破译出一段密码。已知密码可以被看作一个正整数 X ，且满足以下条件：

1. $X \in [39^3, 39^9]$ 。
2. X 的**各个**数位既不为数字 3，也不为数字 9，如数字 18885，10，77421。
3. 对于集合 $S = \{Y \times X \mid Y \in (3,9), Y \in \mathbb{Z}\}$ ，集合内**仅有一个元素**的各个数位仅为数字 3 或者数字 9，如数字 33333，999，39339，99993。而集合内**其他元素**的**各个**数位既不为数字 3，也不为数字 9。
4. 若符合上述条件的数字有多个，则真正的 X 为其中最小的一个。

Input

本题没有输入。

Output

输出符合条件的正整数 X 。

Explanation

数位，指一个数中每一个数字所占的位置。对于数字 397527，它的数位分别为：个位为 7，十位为 2，百位为 5，千位为 7，万位为 9，十万位为 3。

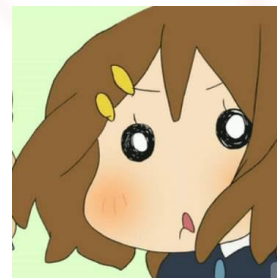
\mathbb{Z} 为整数集合。

This page is intentionally left blank.

此页有意留为空白。

Problem B. The question posed by Hirasawa Yui

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes



今天由于佐和子老师临时有事需要外出，琴吹绫原本准备给佐和子老师的蛋糕多了出来。平泽唯想得到这块多出来的蛋糕，但是轻音部的其他成员为了帮助她通过数学期末考试，给她出了一道题，只有答出这道题才能得到这块多出来的蛋糕，你能帮帮可爱的平泽唯解决这个问题吗？

问题是这样的：给出一个长度为 n 且只包含非负整数的数列 a 和一个数字 m ，找出 k 个长度不超过 m 的互不相同的子数组，使得这 k 个不相同子数组的加和最大。

子数组是数组中一些连续的数字组成的数组，如对于数组 $\{3,6,8,8,10\}$ ，有 $\{3\}$ 、 $\{6,8,8\}$ 、 $\{3,6,8,8,10\}$ 都是它的子数组，但 $\{3,8\}$ 、 $\{6,8,10\}$ 则不是它的子数组。所以每个子数组都可以用取到的原数组的左右端点下标 $\{L, R\}$ 表示，如 $\{6,8,8\}$ 可以用 $[2,4]$ 表示。对于两个子数组 $[L1, R1]$ ， $[L2, R2]$ ，当且仅当 $L1=L2$ 且 $R1=R2$ 时，即取到的左右端点下标完全相同时，认为两个子数组相同。

Input

输入第一行包含 3 个正整数 $n(1 \leq n \leq 10^6)$ ， $m(1 \leq m \leq n)$ ， $k(1 \leq k \leq 2)$ ，分别代表数组长度，子数组最大长度以及需要找的子数组个数。

第二行给出数组 a ，其中数组的第 i 项 $a_i(1 \leq a_i \leq 10^9)$ 。

Output

输出 k 个子数组的加和的最大值。

Examples

standard input	standard output
5 2 1 1 2 8 1 3	10
standard input	standard output
7 3 2 1 2 8 1 3 2 2	23

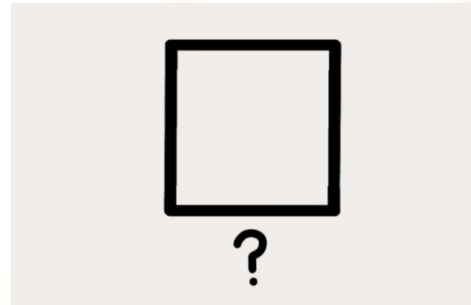
Explanation

第一个样例，取到子数组 $[2,3]$ 时，有子数组最大和 10。

第二个样例，取到子数组 $[2,4]$ 以及子数组 $[3,5]$ 时，有子数组最大加和 $11+12=23$ 。

Problem C. Name

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



本次比赛的出题人是谁？很抱歉，不能直接告诉你。

不过可以给你一点提示：出题人的名字是一个仅由英文大小写字母组成的字符串，且将各字母依次转为 ASCII 码后，再将对应的 ASCII 码依次转化为二进制，可以得到一个仅包含 01 的字符串。

现在，转化后的 01 字符串是已知的，你能否正确将其解码并输出出题人的名字？

Input

输入包含 $q(1 \leq q \leq 100)$ 组测试样例，第一行为测试样例组数 q 。

对于每组测试样例，第一行为 $n, m(1 \leq (n + m) \leq 2000, 7|(n + m))$ ，分别代表数组 z 和数组 o 的长度。第二行为数组 z ，表示所有 0 在 01 字符串中出现的位置下标（1-indexed），第三行为数组 o ，表示所有 1 在 01 字符串中出现的位置下标（1-indexed）。

输入保证合法，每个字母转换后的二进制只有七位，且一定有解。

Output

输出应包含 q 行，每行为出题人名字。

Examples

standard input	standard output
2 9 12 2 4 5 6 7 10 11 17 18 1 3 8 9 12 13 14 15 16 19 20 21 7 14 3 5 7 12 13 16 17 1 2 4 6 8 9 10 11 14 15 18 19 20 21	Pgg jy0

Explanation

第一个例子中，Pgg 的加密过程如下：

P g g

ASCII 码： 80 103 103

01 字符串： 1010000 1100111 1100111

拼凑可得字符串： 101000011001111100111

下标从 1 开始，则有：

z 数组={2,4,5,6,7,10,11,17,18}

o 数组={1,3,8,9,12,13,14,15,16,19,20,21}

（下页还有内容）

附表：ASCII 可显示字符

ASCII可显示字符

二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形
0010 0000	32	20	(空格) (␣)	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	\$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	-	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s
0011 0100	52	34	4	0101 0100	84	54	T	0111 0100	116	74	t
0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u
0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	v
0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	w
0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	x
0011 1001	57	39	9	0101 1001	89	59	Y	0111 1001	121	79	y
0011 1010	58	3A	:	0101 1010	90	5A	Z	0111 1010	122	7A	z
0011 1011	59	3B	;	0101 1011	91	5B	[0111 1011	123	7B	{
0011 1100	60	3C	<	0101 1100	92	5C	\	0111 1100	124	7C	
0011 1101	61	3D	=	0101 1101	93	5D]	0111 1101	125	7D	}
0011 1110	62	3E	>	0101 1110	94	5E	^	0111 1110	126	7E	~
0011 1111	63	3F	?	0101 1111	95	5F					

This page is intentionally left blank.

此页有意留为空白。

Problem D. Black piano key (easy version)

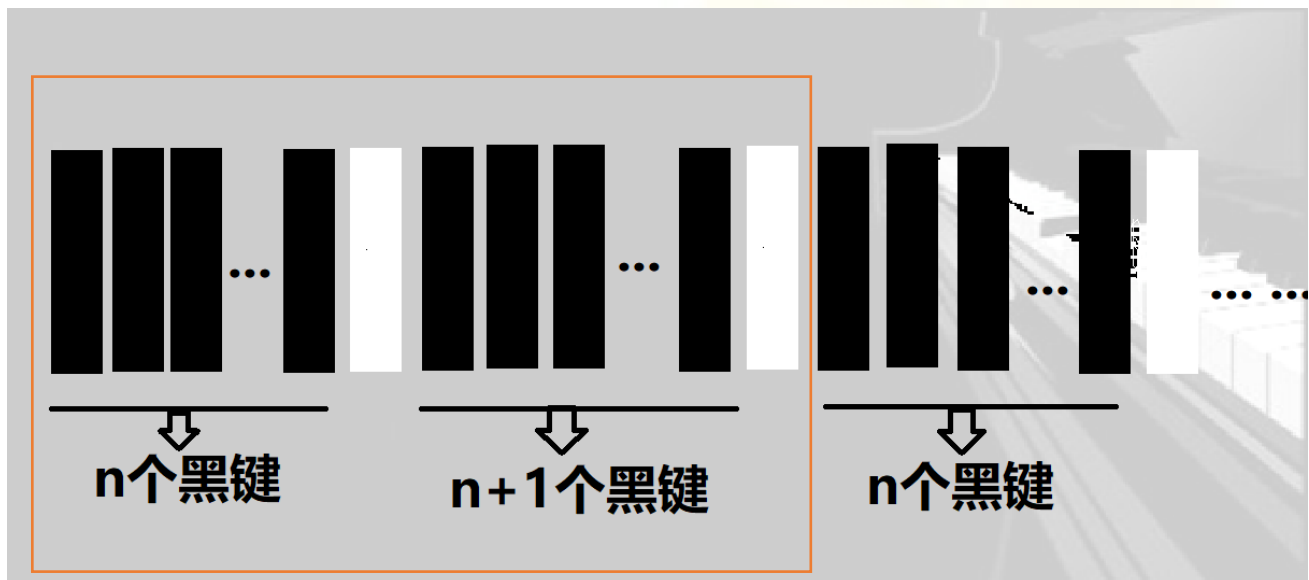
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



注意：此题为 Black piano key 的简单版本，简单版本和困难版本唯一的区别是 n 和 m 的取值范围以及时间限制。

一般的钢琴上，五个黑键和一些白键算做一组，但是天才钢琴少女 Maki 发明了一架每组为 n 个黑键加上 $(n+1)$ 个黑键的钢琴。

这架钢琴一共有 10^{100} 个黑白键，所以钢琴的俯视图可以抽象为下图：



其中橙色方框框选的就是一组，显然一组内包含 $2n+1$ 个黑键和 2 个白键。

现在，给定你两种操作：操作一为将某一组的 2 个白键染黑，操作二是一个询问，询问内容为当前某一个黑键在第几组。你需要回答所有询问。

Input

输入第一行包含两个正整数 $n, m (1 \leq n, m \leq 10^3)$ ，分别表示每一组有 $2n+1$ 个黑键以及 m 个操作。

接下来 m 行，每一行包含一个操作：

1. $0\ i$ ，表示将第 i 组的 2 个白键染黑。 ($1 \leq i \leq 10^9$)
2. $1\ j$ ，表示询问第 j 个黑键所在组。 ($1 \leq j \leq 10^{12}$)

数据保证所有的 i 不重复，即不会重复将同一个白键染黑两次。

Output

对于每个询问，输出一行，表示第 j 个黑键所在组。

Examples

standard input	standard output
2 10	2
1 6	2
1 7	2
0 1	1
1 8	3
1 7	3
1 15	2
0 2	2
1 15	
1 14	
1 9	

Explanation

对于第一个询问和第二个询问，都输出 2，也就是第 6 个和第 7 个黑键都在第 2 组。

接下来将第一组的 2 个白键染黑后，第 8 个黑键就在第 2 组，而第 7 个黑键就在第 1 组了，因为现在第 1 组全是黑键，即 7 个黑键。

之后的操作同理。

Problem E. Enterdawn's hotpot

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



喜大普奔，DLU 终于解封了！于是 enterdawn 决定去海**吃火锅。海**有很多 enterdawn 爱吃的菜，但是 enterdawn 的食量有限，吃下的食物的热量总和最多不能超过 X 。又由于到了月底，enterdawn 的钱也不多了，所以在不超过他的热量摄入上限 X 的前提下，他想花尽可能少的钱摄入尽可能多的热量。所以，请聪明的你编程帮助 enterdawn 计算他最小的花费。

enterdawn 每道菜最多点一份，且 enterdawn 非常节俭，不会浪费，因此会把点的食物吃完（即摄入该食物所有的热量）。

Input

输入第一行为一个正整数 $X(1 \leq X \leq 2 \times 10^3)$ ，表示 enterdawn 能够摄入的热量上限。

第二行为一个正整数 $n(1 \leq n \leq 20)$ ，表示菜品的个数。

接下来 n 行，第 $i(1 \leq i \leq n)$ 行包含两个正整数 $a, b(1 \leq a, b \leq 10^2)$ ，分别表示第 i 个菜品的价格和热量。

Output

输出一个正整数，代表摄入的热量总和尽可能多，且不超过热量上限 X 的前提下的最小花费。

Examples

standard input	standard output
10 5 6 2 3 2 5 5 4 6 6 4	10

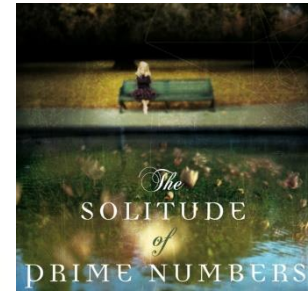
Explanation

点最后两个菜品，可以获得最大的热量总和 10，且没有超过热量上限，此时有花费 $4 + 6 = 10$ ，可以证明在获得热量总和为 10 的前提下，不可能找到花费比 10 更小的方案。

This page is intentionally left blank.
此页有意留为空白。

Problem F. Solitary number

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



“……它们历来都是数学界的宠儿，是数学里神秘的谜团，有关它们的话题总能引发热议，其缘由可能就在于它那孤独又高冷的一面，常常让人可望而不可及……”

小明走在放学回家的路上，脑中还回响着今天数学课上老师的谆谆教诲。回家后他查阅了相关资料，发现课上讲的这些“孤独的数”被广泛应用在密码学，汽车变速箱齿轮上，甚至自然界中也存在这些“孤独的数”。

若一个数是大于 1 的正整数，且只能被 1 和它本身整除，则它就是一个“孤独的数”。现在你的任务是，给你一段闭区间 $[L, R]$ ，请你输出区间内“孤独的数”的数量以及最大的“孤独的数”。

Input

输入第一行包含 2 个正整数，代表闭区间左端点 L 和闭区间右端点 R ($1 \leq L \leq R \leq 10^5$)。

Output

你应该输出两个数字，第一个数字为闭区间 $[L, R]$ 内孤独的数的数量，第二个数字为闭区间 $[L, R]$ 内最大的孤独的数，若给定的闭区间内不存在神奇的数字，请输出“do not exist”（不包含双引号），注意换行。

Examples

standard input	standard output
9 98	21 97
standard input	standard output
998 1008	0 do not exist

Explanation

第一个样例中，显然有孤独的数：11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97，一共有 21 个，且最大的孤独的数为 97。

This page is intentionally left blank.
此页有意留为空白。

Problem G. Black piano key (hard version)

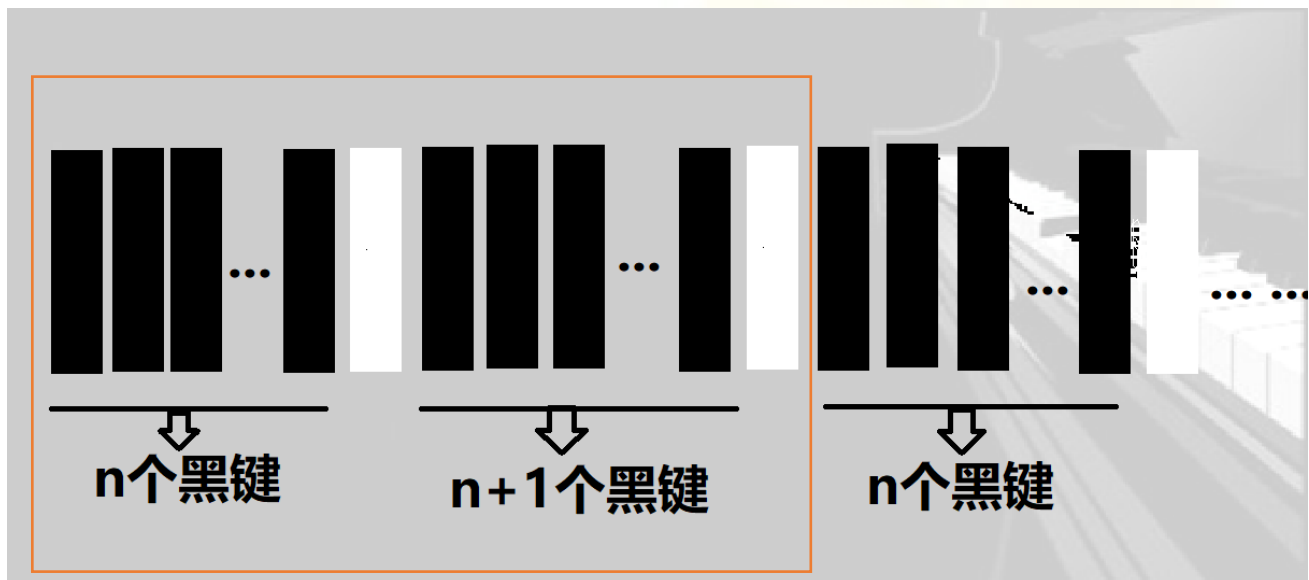
Input file: standard input
 Output file: standard output
 Time limit: 2 seconds
 Memory limit: 256 megabytes



注意：此题为 Black piano key 的困难版本，简单版本和困难版本唯一的区别是 n 和 m 的取值范围以及时间限制。

一般的钢琴上，五个黑键和一些白键算做一组，但是天才钢琴少女 Maki 发明了一架每组为 n 个黑键加上 $(n+1)$ 个黑键的钢琴。

这架钢琴一共有 10^{100} 个黑白键，所以钢琴的俯视图可以抽象为下图：



其中橙色方框框选的就是一组，显然一组内包含 $2n+1$ 个黑键和 2 个白键。

现在，给定你两种操作：操作一为将某一组的 2 个白键染黑，操作二是一个询问，询问内容为当前某一个黑键在第几组。你需要回答所有询问。

Input

输入第一行包含两个正整数 $n, m (1 \leq n, m \leq 10^5)$ ，分别表示每一组有 $2n+1$ 个黑键以及 m 个操作。

接下来 m 行，每一行包含一个操作：

1. $0\ i$ ，表示将第 i 组的 2 个白键染黑。 ($1 \leq i \leq 10^9$)
2. $1\ j$ ，表示询问第 j 个黑键所在组。 ($1 \leq j \leq 10^{12}$)

数据保证所有的 i 不重复，即不会重复将同一个白键染黑两次。

Output

对于每个询问，输出一行，表示第 j 个黑键所在组。

Examples

standard input	standard output
2 10	2
1 6	2
1 7	2
0 1	1
1 8	3
1 7	3
1 15	2
0 2	2
1 15	
1 14	
1 9	

Explanation

对于第一个询问和第二个询问，都输出 2，也就是第 6 个和第 7 个黑键都在第 2 组。

接下来将第一组的 2 个白键染黑后，第 8 个黑键就在第 2 组，而第 7 个黑键就在第 1 组了，因为现在第 1 组全是黑键，即 7 个黑键。

之后的操作同理。

Problem H. The easiest problem in this contest

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes



作为本场比赛最简单的一道题，没有那么多弯弯绕的题面。

在本题中，我们会提供给你一个等差数列的首项 a_1 ，末项 a_n 和公差 d ，你只需要做一件事，就是输出等差数列的和。

忘记了怎么算前 n 项和？没关系，这里有：

$$a_n = a_1 + (n - 1) \times d$$

$$S_n = na_1 + \frac{n(n-1)}{2}d, n \in n^*$$

Input

输入第一行包含 1 个正整数 $t(1 \leq t \leq 10^3)$ ，代表该测试样例有 t 个测试数据。

接下来 t 行，每一行有三个整数 $a_1(-10^3 \leq a_1 \leq 10^3)$ ， $a_n(-10^5 \leq a_n \leq 10^5)$ ， $d(0 < |d| \leq 10^2)$ ，分别表示一个等差数列的首项、末项及公差。

数据保证合法，即有且仅有一个等差数列满足数据输入。

Output

输出应包含 t 行，每行一个整数表示该等差数列的和。

Examples

standard input	standard output
3	25
1 9 2	-30
-10 0 2	5050
1 100 1	

Explanation

第一个数据中，首项为 1，末项为 9，公差为 2，可通过给出的公式得到该等差数列为 $\{1, 3, 5, 7, 9\}$ ，所以和为 $1 + 3 + 5 + 7 + 9 = 25$ ，另外两个数据同理。

This page is intentionally left blank.
此页有意留为空白。

Problem I. ii Genn ki

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



“在大连大学，三年级学生说听不见就是听不见。”

当你的同学被高年级老大哥训斥后，你发现事情不妙——因为你忘了你的名字。但庆幸的是你记得你的名字是两个整数 a 和 b 的和。

Input

输入第一行包含一个整数 N ($1 \leq N \leq 100$)，表示一共有 N 组测试数据。

接下来 N 行，每行包含两个整数 a, b ($|a| \leq 1000, |b| \leq 1000$)。

Output

输出应包含 N 行，每行输出一个整数表示 a 与 b 的和。

Examples

standard input	standard output
2	6
1 5	30
10 20	