

题目名称	魔法石	数数	gcd	逆序对
输入文件名	<a href="#">magic.in</a>	<a href="#">count.in</a>	<a href="#">gcd.in</a>	<a href="#">inverse.in</a>
输出文件名	magic.out	count.out	gcd.out	inverse.out
每个测试点时限	1 s	1 s	1 s	1 s
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
内存限制	256MB	256MB	256MB	256MB
是否有部分分	有	有	有	有
题目类型	传统	传统	传统	传统

## Problem A. 魔法石 (magic.c/cpp/pas)

Input file: [magic.in](#)

Output file: magic.out

Time limit: 1 second

Memory limit: 256 megabytes

在一个远古的数字王国里，勇敢的探险者 Luke 正在寻找传说中的魔法石。这些魔法石的特性被一个古老的预言所描述，预言中提到，石头的属性与三个神秘正整数  $x$ ,  $y$ , 和  $z$  密切相关。为了获得这些魔法石，Luke 必须找到三个正整数  $a$ ,  $b$ , 和  $c$ , 使得以下条件都得到满足：

1.  $x$  是  $a$  和  $b$  的最大值，即  $x = \max(a, b)$ 。
2.  $y$  是  $a$  和  $c$  的最大值，即  $y = \max(a, c)$ 。
3.  $z$  是  $b$  和  $c$  的最大值，即  $z = \max(b, c)$ 。

然而，预言还提到，Luke 必须解决一道复杂的谜题才能完成寻找：

1. 对于每对数值  $(a, b)$ ,  $(a, c)$ , 和  $(b, c)$ , 计算它们的最小值和最大值，得到新的值对  $(u, v)$ ,  $(w, x)$ , 和  $(y, z)$ , 其中  $u = \min(a, b)$ ,  $v = \max(a, b)$ ,  $w = \min(a, c)$ ,  $x = \max(a, c)$ ,  $y = \min(b, c)$ ,  $z = \max(b, c)$ 。
2. 确保所有这些新计算的最小值和最大值都不大于给定的  $x$ ,  $y$ , 和  $z$  中的相应值。

帮助 Luke 解开这些谜题，将使他获得通往魔法石的最终路径。他必须在规定的时间内找出所有满足条件的正整数  $a$ ,  $b$ , 和  $c$  的组合，才能成功完成他的探险任务。

### Input

一行包含三个整数  $x$ 、 $y$  和  $z$  ( $1 \leq x, y, z \leq 10^9$ )。

### Output

- 如果不存在解决方案，则在输出的唯一一行中显示“NO”；

- 或在第一行中显示“YES”，在第二行中显示任何有效的三元组正整数  $a$ 、 $b$  和  $c$  ( $1 \leq a, b, c \leq 10^9$ )。您可以按任意顺序打印  $a$ 、 $b$  和  $c$ 。

## Examples

### 【样例 1 输入】

3 2 3

### 【样例 1 输出】

YES  
3 2 1

### 【样例 2 输入】

1 1000000000 1000000000

### 【样例 2 输出】

YES  
1 1 1000000000

### 【样例 3 输入】

10 30 20

### 【样例 3 输出】

NO  
1 1 1000000000

## Notes

对于60%的数据,  $1 \leq a, b, c \leq 10^9$

对于100%的数据,  $1 \leq a, b, c \leq 10^{18}$

## Problem B. 数数 (count.c/cpp/pas)

---

**Input file:**        **count.in**

**Output file:**      count.out

**Time limit:**        1 second

**Memory limit:**    256 megabytes

在数字王国中，著名的算法优化专家 Luke 正在研究一种新的排列优化问题。他最近生成了一个从 1 到  $n$  的随机排列  $P$ 。在这个排列中，Luke 定义了一个新的度量标准：对于区间  $[l, r]$ （其中  $1 \leq l \leq r \leq n$ ），该区间的价值被定义为区间内所有元素差的最大值，即：

$$\text{价值} = \max(P[i] - P[j]), \text{ 其中 } l \leq i, j \leq r$$

Luke 想要深入了解这个度量标准，以便更好地优化他的算法。他需要确定所有区间的价值的总和为多少。

## Input

第一行一个整数  $T$ ，表示数据组数

对于每一组数据：

第一行一个数  $n$

第二行  $n$  个数  $a_i$ ，表示一个  $1 \sim n$  的随机的排列

## Output

对于每组数据输出一个数，表示答案

## Examples

### 【样例 1 输入】

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

### 【样例 1 输出】

```
14
```

## Notes

对于60%的数据， $1 \leq n \leq 1000$

对于100%的数据， $1 \leq n \leq 100,000$

## Problem C. gcd (gcd.c/cpp/pas)

---

Input file: [gcd.in](#)

---

Output file: gcd.out

---

Time limit: 1 second

---

Memory limit: 256 megabytes

Luke 是一名充满好奇心的数学探险家。他最近迷上了一种神秘的数字游戏，游戏规则很简单：在一片数字大陆上，Luke 需要找到特定的数字对。每次，他都要选择两个数字  $a$  和  $b$ ，并且要求它们满足一些奇特的关系。

在这片数字大陆上，有一个强大的神秘力量，它就是“最大公约数” (gcd)，以及一个神秘的运算符“异或” (xor)。Luke 的任务是找到一对  $a$  和  $b$  满足  $\gcd(a, b) = a \text{ xor } b$ 。

然而，这并不是那么简单！Luke 发现这对数字必须位于  $[1, n]$  之间，且他只能找出无序的数字对，也就是说  $(a, b)$  和  $(b, a)$  是相同的。

现在，Luke 需要你的帮助，来找出在给定的数字范围内有多少对符合要求的数字对。快来帮助 Luke 一起解开这个谜题吧！

### Input

输入共一行，一个整数  $n$ 。

### Output

输出一行一个整数，即答案

### Examples

【样例 1 输入】

3

【样例 1 输出】

1

### Notes

对于30%的数据， $n \leq 10^3$

对于60%的数据， $n \leq 10^5$

对于100%的数据， $n \leq 10^7$

## Problem D. 逆序对 (inverse.c/cpp/pas)

---

Input file: [inverse.in](#)

---

Output file: inverse.out

---

Time limit: 1 second

---

Memory limit: 256 megabytes

Luke 发现人们的幸福感取决于一个长度为 $n$ 的数字序列。

她有 $m$ 个机会，每个机会都允许她交换序列中的两个数字，但也可以选择交换。

一个人的幸福感由序列中的逆序对数量决定，其中一个逆序对是指满足 $i < j \leq n$ 且 $a_i > a_j$ 的数字对。

Luke 对于这 $m$ 个机会的所有可能操作（每个机会都有选择交换或不交换的两种方式）非常好奇。

她想知道所有这些操作方式中，总的幸福感值之和是多少。由于这个值可能非常大，请你输出其对 1000000007 取模后的结果。

### Input

第一行两个正整数  $n, m$ 。

接下来  $n$  个数表示  $a_i$ 。

接下来  $m$  个数对  $x, y$ ，第  $i$  个数对表示这个操作可以交换  $a_x$  和  $a_y$

### Output

输出一行一个整数表示答案。

### Examples

#### 【样例 1 输入】

```
3 2
1 2 3
1 2
1 3
```

#### 【样例 1 输出】

```
6
```

#### 【样例 1 解释】

抓住两个机会，最终序列为(3,1,2)，逆序对数量为 2

只抓住第一个机会，最终序列为(2,1,3)，逆序对数量为 1

只抓住第二个机会，最终序列为(3,2,1)，逆序对数量为 3

两个机会都不抓住，最终序列为(1,2,3)，逆序对数量为 0

$$2 + 1 + 3 + 0 = 6$$

## Notes

对于20%的数据， $n, m \leq 10$

对于50%的数据， $n, m \leq 100$

对于100%的数据， $n, m \leq 1000$