2025~26 算法入门 C++

KGOI-J（初级组）

时间：2026 年 02 月 27 日 17:00 ~ 19:00

题目名称	歌曲	公路	序列求和	代码重测
题目类型	传统型	传统型	传统型	传统型
目录	song	road	sum	rejudge
输入文件名	stdin	stdin	stdin	stdin
输出文件名	stdout	stdout	stdout	stdout
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512MiB	512MiB	512MiB	512MiB
测试点数目	10	10	20	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	song.cpp	road.cpp	sum.cpp	rejudge.cpp

注意事项：

1. 本比赛使用 OI 赛制，选手提交后不会提供实时反馈，且对于每题，只会对该题的最后一次提交评分。
2. 时限为 120 分钟。每题满分为 100 分，总分 400 分。
3. C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
4. 若无特殊说明，结果的比较方式为全文比较 (即过滤文末空格及文末行)。
5. 选手需要把源程序提交到 Google Classroom 的指定教室中。
6. 采用洛谷的配置作为评测配置。上述时限以此配置为准。
7. **注意：由于洛谷环境下的换行符为 \r\n，如需读取换行符应连续读取两个字符，例如执行两次 getchar()。**
8. 结束后，在正常情况下系统会显示 0 分，这不是选手的成绩。
9. 选手的成绩、题面、样例数据及真实数据将于比赛结束一周内公布于 Google Classroom。

歌曲（song）

【题目描述】

有一天，小 Z 想到了一首奇怪的歌曲，这首歌曲包含 n 个歌词。在这里，每个歌词可以被看成一个字符串。

小 Z 随机生成了 n 个非负整数 $a[1], a[2], a[3], \ldots, a[n]$ 和 m 个长度不大于 5 且仅由英文字母组成的字符串 $s[1], s[2], s[3], \ldots, s[m]$。记歌曲的第 i 个歌词为 $l[i]$，它需要符合以下规定：

- $x[i] = a[i] \bmod m + 1$
- $l[i] = s[x[i]]$

小 Z 希望用这样的方法完成歌曲，但由于 n, m 很大，小 Z 不喜欢进行大量的计算。于是，他把这个任务交给你。

【输入格式】

從标准输入 (stdin) 读入数据。

输入的第一行包含两个正整数 n, m，以空格分隔。

第二行包含 n 个非负整数 $a[1], a[2], a[3], \ldots, a[n]$，以空格分隔。

第三行包含 m 个字符串 $s[1], s[2], s[3], \ldots, s[m]$，以空格分隔。

【输出格式】

输出到标准输出 (stdout) 中。

输出一行，包含 n 个字符串 $l[1], l[2], l[3], \ldots, l[n]$，以空格分隔。

【样例 1 输入】

```
5 3
2 3 1 0 4
is happy Tom
```

【样例 1 输出】

```
Tom is happy is happy
```

【样例 1 解释】

$x[1] = a[1] \bmod m + 1 = 2 \bmod 3 + 1 = 3$，故 $l[1] = s[x[1]] = s[3]$，即 Tom。

$x[2] = a[2] \bmod m + 1 = 3 \bmod 3 + 1 = 1$，故 $l[2] = s[x[2]] = s[1]$，即 is。

$x[3] = a[3] \bmod m + 1 = 1 \bmod 3 + 1 = 2$，故 $l[3] = s[x[3]] = s[2]$，即 happy。

$x[4] = a[4] \bmod m + 1 = 0 \bmod 3 + 1 = 1$，故 $l[4] = s[x[4]] = s[1]$，即 is。

$x[5] = a[5] \bmod m + 1 = 4 \bmod 3 + 1 = 2$，故 $l[5] = s[x[5]] = s[2]$，即 happy。

因此，应输出 Tom is happy is happy。

【样例 2】

见 song/song2.in 与 song/song2.ans。

这个样例满足测试点 1 ~ 2 的约束条件。

【样例 3】

见 song/song3.in 与 song/song3.ans。

这个样例满足测试点 3 ~ 6 的约束条件。

【样例 4】

见 song/song4.in 与 song/song4.ans。

这个样例满足测试点 7 ~ 10 约束条件。

【数据范围】

对于所有测试数据，保证：

- $n \leq 10^6$
- $m \leq 10^6$
- 对于所有满足 $1 \leq i \leq n$ 的整数 i，$a[i] \leq 10^9$。
- 对于所有满足 $1 \leq i \leq m$ 的整数 i，$s[i]$ 的长度不超过 5。

测试点	$n \leq$	$m \leq$	特殊性质
1 ~ 2	1000	1	无
3 ~ 6		10	
7 ~ 10	10^6	10^6	

公路（road）

【题目描述】

小 E 驾驶着汽车通往一条笔直的公路。

公路上有 n 个休息站，而每两个休息站之间会有一段距离。具体而言，对于所有满足 $1 \leq i < n$ 的整数 i，第 i 个和第 $i+1$ 个休息站的距离为 $d[i]$ 公里。对于每一段距离，小 E 的汽车均以恒定的速度 v（公里/小时）匀速直线运动。

小 E 只要经过休息站就会上前休息。在第 i 个休息站中，他必须休息 $t[i]$ 小时才有力气继续驾驶汽车。而第 n 个休息站是终点。

小 E 想知道，他总共需要多少时间（以小时为单位）才能抵达终点。

【输入格式】

從标准输入（stdin）读入数据。

输入的第一行包含两个正整数 n, v，以空格分隔。

第二行包含 $n-1$ 个正整数 $d[1], d[2], d[3], \ldots, d[n-1]$，以空格分隔。

第三行包含 $n-1$ 个正整数 $t[1], t[2], t[3], \ldots, t[n-1]$，以空格分隔。

【输出格式】

输出到标准输出（stdout）中。

输出一行，包含两个正整数 x, y，以空格分隔。其中，x, y 满足答案为 x/y 且该分数为最简分数。

【样例 1 输入】

```
5 4
6 3 5 4
2 7 3 8
```

【样例 1 输出】

```
49 2
```

【样例 1 解释】

在第一个休息站休息需历时 2 小时，通过第一段道路需历时 6/4 小时。

在第二个休息站休息需历时 7 小时，通过第二段道路需历时 3/4 小时。

在第三个休息站休息需历时 3 小时，通过第三段道路需历时 5/4 小时。

在第四个休息站休息需历时 8 小时，通过第四段道路需历时 $4/4$ 小时。

最终到达终点，共历时 $2+6/4+7+3/4+3+5/4+8+4/4=49/2$ 小时。

【样例 2】

见 road/road2.in 与 road/road2.ans。

这个样例满足测试点 1 的约束条件。

【样例 3】

见 road/road3.in 与 road/road3.ans。

这个样例满足测试点 2 的约束条件。

【样例 4】

见 road/road4.in 与 road/road4.ans。

这个样例满足测试点 3 ~ 5 的约束条件。

【样例 5】

见 road/road5.in 与 road/road5.ans。

这个样例满足测试点 6 ~ 7 的约束条件。

【样例 6】

见 road/road6.in 与 road/road6.ans。

这个样例满足测试点 8 的约束条件。

【样例 7】

见 road/road7.in 与 road/road7.ans。

这个样例满足测试点 9 ~ 10 的约束条件。

【数据范围】

对于所有测试数据，保证：

- $n \leq 10^6$
- $v \leq 10^6$
- 对于所有满足 $1 \leq i < n$ 的整数 i，$d[i] \leq 10^6$。
- 对于所有满足 $1 \leq i < n$ 的整数 i，$t[i] \leq 10^6$。

测试点	$n \leq$	$v \leq$	$d[i] \leq$	$t[i] \leq$	特殊性质
1	1000	1	1000	1000	无
2		2			
3 ~ 5		1000			
6 ~ 7	10^6	2			
8		10^6			
9 ~ 10			10^6	10^6	

序列求和 (sum)

【题目描述】

有一天，老师分别给了小 A 和小 B 各一张纸，这两张纸上面的内容是相同的，都包含一个长度为 n 的整数序列 a。

老师让它们在序列当中选择 m 个元素，并把它们修改为 $a[1], a[2], a[3], \ldots, a[n]$ 中的其中一个。其中，每一次改值的操作都是独立的，不受其他操作影响。

老师想让他们在修改后的序列中，选取当中的 k 个元素并求出它们的和。小 A 希望这个答案尽可能地大，小 B 则希望这个答案尽可能地小。

小 A 和小 B 都是十分聪明的学生。他们可以找出最优方案并求出他们想要的答案。现在作为旁观者的你希望知道，小 A 和小 B 最后求出的答案分别是什么。

【输入格式】

從标准输入 (stdin) 读入数据。

输入的第一行包含三个正整数 n, m, k，以空格分隔。

第二行包含 n 个整数 $a[1], a[2], a[3], \ldots, a[n]$，以空格分隔。

【输出格式】

输出到标准输出 (stdout) 中。

输出两个整数，分别为小 A 和小 B 最后求出的答案，以空格分隔。

【样例 1 输入】

```
5 2 3
2 3 1 0 -4
```

【样例 1 输出】

```
9 -12
```

【样例 2】

见 sum/sum2.in 与 sum/sum2.ans。

这个样例满足测试点 1 ~ 4 的约束条件。

【样例 3】

见 sum/sum3.in 与 sum/sum3.ans。

这个样例满足测试点 5 ~ 6 的约束条件。

【样例 4】

见 sum/sum4.in 与 sum/sum4.ans。

这个样例满足测试点 7 ~ 10 的约束条件。

【样例 5】

见 sum/sum5.in 与 sum/sum5.ans。

这个样例满足测试点 13 ~ 14 的约束条件。

【样例 6】

见 sum/sum6.in 与 sum/sum6.ans。

这个样例满足测试点 15 ~ 20 的约束条件。

【数据范围】

对于所有测试数据，保证：

- $n \leq 10^6$
- $m \leq n$
- $k \leq n$
- 对于所有满足 $1 \leq i \leq n$ 的整数 i，$-10^9 \leq a[i] \leq 10^9$。

测试点	$n \leq$	特殊性质
1 ~ 4	15	无
5 ~ 6	1000	B
7 ~ 10		无
11 ~ 12	10^6	A
13 ~ 14		B
15 ~ 20		无

特殊性质 A：$k = n$。

特殊性质 B：$m \geq k - 1$。

代码重测 (rejudge)

【题目描述】

小 S 举办了一场 OI 比赛，这场比赛有 n 位参赛者，编号分别为 $1, 2, 3, \ldots, n$。比赛结束后，编号为 i 的参赛者的分数为 $a[i]$。

小 S 认为，分数高于一定标准分数 r 的参赛者，可以直接晋级；而分数低于一定标准分数 l 的参赛者，会被直接淘汰。基于某些原因，小 S 认为其他参赛者的代码需要进行重测。小 S 想知道有哪些参赛者的代码需要重测。

由于小 S 会随时改变主意，因为他会询问你 q 次，对于第 i 次询问：

- 他会给你两个标准分数 $l[i], r[i]$，分数高于 $r[i]$ 的参赛者会直接晋级，而分数低于 $l[i]$ 的参赛者会直接淘汰。
- 小 S 关心的是需重测的参赛者的编号之和是什么。你需要回答这个答案。

【输入格式】

從标准输入 (stdin) 读入数据。

输入的第一行包含两个正整数 n, q，以空格分隔。

第二行包含 n 个正整数 $a[1], a[2], a[3], \ldots, a[n]$，以空格分隔。

接下来 q 行，第 i 行表示第 i 次询问：

- 包含两个正整数 $l[i], r[i]$，以空格分隔。

【输出格式】

输出到标准输出 (stdout) 中。

为了减少输出量，记第 i 次询问的答案为 $ans[i]$，你只需要输出 $(1 \times ans[1]) \oplus (2 \times ans[2]) \oplus (3 \times ans[3]) \oplus \ldots \oplus (q \times ans[q]) \bmod 2^{64}$ 的值。其中 $\oplus$ 表示二进制按位异或运算。

【样例 1 输入】

```
8 2
72 93 2 82 23 10 45 33
23 45
17 92
```

【样例 1 输出】

```
38
```

【样例 1 解释】

对于第一次询问：编号为 $5,7,8$ 的参赛者需重测，编号之和为 $5+7+8=20$。

对于第二次询问：编号为 $1,4,5,7,8$ 的参赛者需重测，编号之和为 $1+4+5+7+8=25$。

应输出 $(1\times 20)\oplus(2\times 25)\bmod 2^{64}=38$。

【样例 2】

见 `rejudge/rejudge2.in` 与 `rejudge/rejudge2.ans`。

这个样例满足测试点 `1 ~ 2` 的约束条件。

【样例 3】

见 `rejudge/rejudge3.in` 与 `rejudge/rejudge3.ans`。

这个样例满足测试点 `3 ~ 4` 的约束条件。

【样例 4】

见 `rejudge/rejudge4.in` 与 `rejudge/rejudge4.ans`。

这个样例满足测试点 `5 ~ 6` 的约束条件。

【样例 5】

见 `rejudge/rejudge5.in` 与 `rejudge/rejudge5.ans`。

这个样例满足测试点 `9 ~ 10` 的约束条件。

【数据范围】

对于所有测试数据，保证：

- $n\le 10^6$
- $q\le 10^6$
- 对于所有满足 $1\le i\le n$ 的整数 i，$a[i]\le 10^9$。
- 对于所有满足 $1\le i\le q$ 的整数 i，$l[i]\le r[i]\le 10^9$。

测试点	$n \leq$	$q \leq$	$a[i] \leq$	特殊性质
1 ~ 2	1000	1000	1000	无
3 ~ 4	10^6	10^6	10^6	A
5 ~ 6				无
7 ~ 8			10^9	B
9 ~ 10				无

特殊性质 A：对于所有满足 $1 \leq i \leq q$ 的整数 i，$l[i] = 1$。

特殊性质 B：对于所有满足 $1 \leq i < n$ 的整数 i，$a[i] \leq a[i+1]$。