

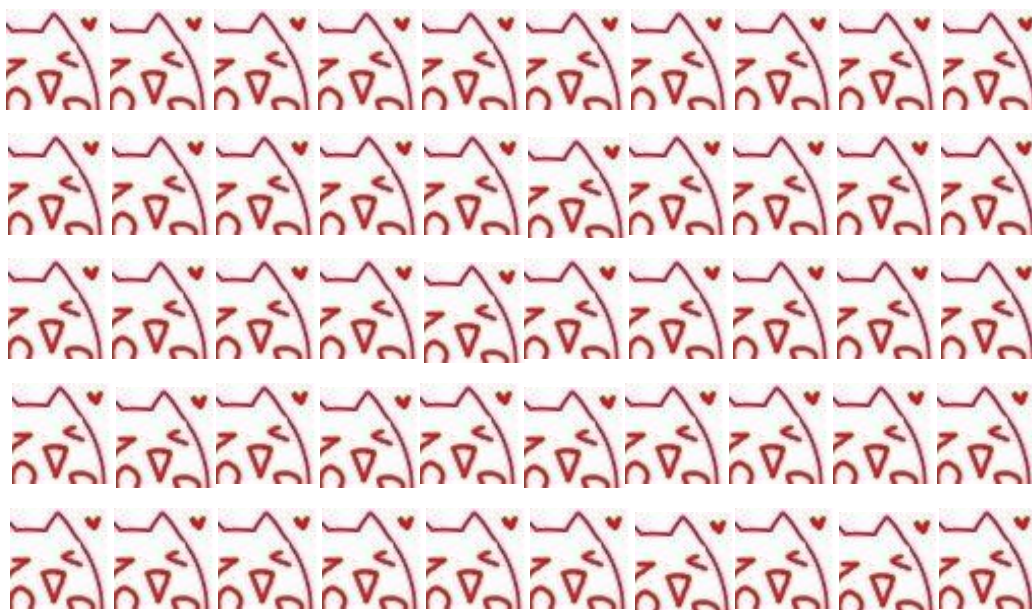
(Y)Noi2019(?) 模拟赛

一、题目概览

中文题目名称	末日时在做什么	有没有空	可以来拯救吗
英文题目名称	end	busy	save
可执行文件名	end	busy	save
输入文件名	end.in	busy.in	save.in
输出文件名	end.out	busy.out	save.out
时间限制	1s	1s	4s
空间限制	512MB	128MB	64-512MB
测试点数目	10	subtask	subtask
测试点分值	10	subtask	subtask
题目类型	传统	传统	传统
比较方式	全文比较	全文比较	全文比较
是否有部分分	是	是	是

二、注意事项:

1. 文件名（程序名和输入输出文件名）必须使用小写。
2. C/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。
3. 评测环境为 Windows，使用 `lemon` 进行评测。
4. 开启 O2 优化，栈空间开大至 256M。
5. AK 了不要 D 出题人，没 AK 也不要 D 出题人。
6. 我请 AK 的人吃饭，200 以内



末日时在做什么

【题目描述】

在 0 到 $N-1$ 的整数里，选择 n 个整数，使得对于每个二进制位，这 n 个整数恰好有偶数个这一位为 1，求方案数。

由于 N 很大，方便起见这里给出 N 的二进制表示， N 由字符串 S 重复 k 次得到。

【输入数据】

第一行两个整数 n, k ，第二行一个 01 串 S ，保证 S 的第一个字符为 1。

【输出数据】

一行，一个非负整数，表示答案对 1000000007 取模的结果。

【样例输入】

```
3 2
10
```

【样例输出】

```
8
```

【样例解释】

给出的 N 为二进制的 1010，即十进制的 10，合法的方案如下：

(1,2,3),(1,4,5),(1,6,7),(1,8,9),(2,4,6),(2,5,7),(3,4,7),(3,5,6)。

【数据范围】

共 10 个测试点，每个测试点的性质如下：

测试点 1: $n=3$, $k=1$, S 的长度 ≤ 5

测试点 2: $n=7$, $k=1$, S 的长度 ≤ 5

测试点 3,4,5: $3 \leq n \leq 7$, $k=1$, S 的长度 ≤ 10

测试点 6,7: $3 \leq n \leq 7$, $1 \leq k \leq 10$, S 的长度 ≤ 50

测试点 8,9: $3 \leq n \leq 7$, $1 \leq k \leq 100000$, S 的长度 ≤ 50

测试点 10: $n=7$, $1 \leq k \leq 100000$, S 的长度 ≤ 10000

对于 100% 的数据，有 $3 \leq n \leq 7$, $1 \leq S$ 的长度 ≤ 10000 , $1 \leq k \leq 100000$

有没有空

【题目描述】

给你一个长为 n 的序列 a

你需要实现 m 个操作，操作有两种：

1. 把序列中所有值为 x 的数的值变成 y
2. 找出一个位置 i 满足 $a_i == x$ ，找出一个位置 j 满足 $a_j == y$ ，使得 $|i-j|$ 最小，并输出 $|i-j|$

【输入数据】

第一行两个数 n, m

之后一行 n 个数，表示序列 a

之后 m 行，每行三个数 $opt\ x\ y$

如果 opt 为 1，代表把序列中所有值为 x 位置的值变成 y

如果 opt 为 2，代表找出一个位置 i 满足 $a_i == x$ ，找出一个位置 j 满足 $a_j == y$ ，使得 $|i-j|$ 最小，并输出 $|i-j|$

本题强制在线，输入的 x 和 y 需要 xor 上一次询问的答案

对于第一次询问，上一次答案为 0

如果上一次输出 Chtholly，xor 时认为上一次的答案为 0

【输出数据】

对于每个 2 操作，输出一行一个数表示答案

如果无法找出满足题意的 i, j ，则输出 Chtholly

【样例输入】

```
5 5
1 2 2 4 4
2 3 3
2 2 4
1 3 2
1 5 5
2 2 5
```

【样例输出】

```
Chtholly
```

1
1

【数据范围】

Subtask1: 20% $1 \leq n, m \leq 1000$

Subtask2: 30% 只有操作 2

Subtask3: 50% 无特殊性质

对于 100% 的数据, $0 \leq$ 出现的所有数在解密后 ≤ 100000

可以来拯救吗

【题目描述】

维护一个由 n 个非负整数构成的序列 $a[1], a[2], a[3] \dots a[n]$ ，支持三种操作：

1. 给定区间 $[l, r]$ ，将对应区间内的数异或上 x
2. 给定区间 $[l, r]$ ，将对应区间内的数从小到大排序
3. 给定区间 $[l, r]$ ，求区间内的数的异或和

【输入数据】

第一行 n, m 两个整数

第二行 n 个整数 $a[i]$ ，表示原序列

接下来 m 行，每行三个整数 op, l, r ($op=2, 3$) 或四个整数 op, l, r, x ($op=1$) 表示对应的操作

【输出数据】

对每个操作 3，输出一行，表示对应的答案

【样例输入】

```
5 3
1 4 2 8 3
2 1 3
1 2 4 5
3 1 2
```

【样例输出】

```
6
```

【样例解释】

原序列为 1 4 2 8 3

将区间 $[1, 3]$ 排序，得到 [1 2 4] 8 3

将区间 $[2, 4]$ 与 5 按位取异或， $2 \text{ xor } 5 = 7$ ， $4 \text{ xor } 5 = 1$ ， $8 \text{ xor } 5 = 13$ ，得到 1 [7 1 13] 3

查询区间 $[1, 2]$ 的异或和， $[1 \ 7] \ 1 \ 13 \ 3$ ， $1 \text{ xor } 7 = 6$

【数据范围】

Subtask1: 20% $1 \leq n, m \leq 1000$

Subtask2: 20% $1 \leq n, m \leq 10000$

Subtask3: 20% 只有操作 1, 3

Subtask4: 20% 空间限制为 512MB

Subtask5: 20% 无特殊性质

对于 100%的数据，满足 $1 \leq n, m \leq 100000$, $0 \leq a[i], x < 1000000000$