

在小美的国家，任何一篇由英文字母组成的文章中，如果大小写字母的数量不相同会被认为文章不优雅。

现在，小美写了一篇文章，并且交给小团来修改。小美希望文章中的大小写字母数量相同，所以她想让小团帮她把某些小写字母改成对应的大写字母，或者把某些大写字母改成对应的小写字母，使得文章变得优雅。

小美给出的文章一定是由偶数长度组成的，她想知道最少修改多少个字母，才能让文章优雅。

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String paper = br.readLine().trim();
        int n = paper.length();
        int countUpper = 0;
        for(int i = 0; i < n; i++){
            char c = paper.charAt(i);
            if(c >= 65 && c <= 90) countUpper ++;
        }
        System.out.println(Math.abs(2*countUpper - n)/2);
    }
}
```

小团有一个序列 a ，下标从 1 开始直到 n ，分别为 a_1, a_2, \dots, a_n 。现在，小团定义了以下式子：

现在小团想让小美回答

$f(a, k)$ 的值

其中， \oplus 代表异或运算

请你帮助小美。

小提示： $f(a, k) = a_1 \oplus a_2 \oplus \dots \oplus a_k$

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine().trim());
        String[] strA = br.readLine().trim().split(" ");
        int[] a = new int[n + 1];
```

```

int[] multi = new int[n + 1]; // multi[i]表示  $0^1 2^2 \dots^i$ 
int res = 0;
// 先把所有的 a[i]进行异或
for(int i = 1; i <= n; i++) {
    a[i] = Integer.parseInt(strA[i - 1]);
    res ^= a[i];
    multi[i] = multi[i - 1]^i;
}
for(int i = 1; i <= n; i++){
    if((n/i) % 2 == 0)
        res ^= multi[n % i];
    else
        res ^= multi[n % i] ^ multi[i - 1];
}
System.out.println(res);
}
}

```

A 国和 B 国正在打仗，他们的目的是 n 块土地。现在，A 国和 B 国暂时休战，为了能合理分配这一些土地，AB 国开始协商。

A 国希望得到这 n 块土地中的 p 块土地，B 国希望得到这 n 块土地中的 q 块土地。每个国家都将自己希望得到的土地编号告诉了小团和小美——两位战争调和人。你需要帮小团和小美计算，有多少块土地是只有 A 国想要的，有多少块土地是只有 B 国想要的，有多少块土地是两个国家都想要的。

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.util.HashSet;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String[] params = br.readLine().trim().split(" ");
        int n = Integer.parseInt(params[0]);
        int p = Integer.parseInt(params[1]);
        int q = Integer.parseInt(params[2]);
        HashSet<Integer> wantedA = new HashSet<>();
        HashSet<Integer> wantedB = new HashSet<>();
        params = br.readLine().trim().split(" ");
        for(int i = 0; i < params.length; i++)
            wantedA.add(Integer.parseInt(params[i]));
    }
}

```

```

// 计算交集的数量
int commonCount = 0;
params = br.readLine().trim().split(" ");
for(int i = 0; i < params.length; i++){
    wantedB.add(Integer.parseInt(params[i]));
    if(wantedA.contains(Integer.parseInt(params[i]))) commonCount ++;
}
System.out.println(wantedA.size() - commonCount + " " + (wantedB.size() - commonCount) + " " + commonCount);
}
}

```

在小团的公司中，有 n 位员工。除了最高领导——小团外，每位员工有且仅有一位直接领导。所以，公司内从属关系可以看成一棵树。

现在，公司接到一个项目，需要重新划分这 n 位员工的从属关系。新的划分描述如下：

- 1.每个人要么没有下属，要么有至少两个直接下属（即至少有两人的直接领导为这个人）
- 2.第 i 个人的下属（包括自己）有恰好 i 个。

请注意，直接下属和下属（包括自己）可分别看做树上点的"儿子"和"子树"。

请问是否存在这么一种关系？注意，输入不会给出最高领导的编号。