

题目描述：

对一个数据 a 进行分类，分类方法为：此数据 a （四个字节大小）的四个字节相加对一个给定的值 b 取模，如果得到的结果小于一个给定的值 c ，则数据 a 为有效类型，其类型为取模的值；如果得到的结果大于或者等于 c ，则数据 a 为无效类型。

比如一个数据 $a=0x01010101$ ， $b=3$ ，按照分类方法计算

$(0x01+0x01+0x01+0x01)\%3=1$ ，所以如果 $c=2$ ，则此 a 为有效类型，其类型为 1 ，

如果 $c=1$ ，则此 a 为无效类型；

又比如一个数据 $a=0x01010103$ ， $b=3$ ，按照分类方法计算

$(0x01+0x01+0x01+0x03)\%3=0$ ，所以如果 $c=2$ ，则此 a 为有效类型，其类型为 0 ，

如果 $c=0$ ，则此 a 为无效类型。

输入 12 个数据，第一个数据为 c ，第二个数据为 b ，剩余 10 个数据为需要分类的数据，请找到有效类型中包含数据最多的类型，并输出该类型含有多少个数据。

输入描述：

输入 12 个数据，用空格分隔，第一个数据为 c ，第二个数据为 b ，剩余 10 个数据为需要分类的数据。

输出描述：

输出最多数据的有效类型有多少个数据。

示例 1

输入：

3 4 256 257 258 259 260 261 262 263 264 265

输出：

3

说明：

10 个数据 4 个字节相加后的结果分别为 1 2 3 4 5 6 7 8 9 10，故对 4 取模的结果为 1 2 3 0 1 2 3 0 1 2，c 为 3，所以 0 1 2 都是有效类型，类型为 1 和 2 的有 3 个数据，类型为 0 的只有 2 个数据，故输出 3

示例 2

输入：

1 4 256 257 258 259 260 261 262 263 264 265

输出：

2

说明：

10 个数据 4 个字节相加后的结果分别为 1 2 3 4 5 6 7 8 9 10，故对 4 取模的结果为 1 2 3 0 1 2 3 0 1 2，c 为 1，所以只有 0 是有效类型，类型为 0 的有 2 个数据，故输出 2

```
// 251658240 15728640 983040 65536 4096 256 16 1
```

```
import java.util.Scanner;
import java.util.HashMap;
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int c = in.nextInt();
        int b = in.nextInt();
        int[] nums = new int[12];
        int i = 0;
        HashMap<Integer, Integer> map = new HashMap<>();
```

while (in.hasNextInt()) { // 注意，如果输入是多个测试用例，请通过 while 循环处理多个测试用例

```
        int num = in.nextInt();
        String str = Integer.toBinaryString(num);
        nums[i] = num/251658240*16;
        num%=251658240;
        nums[i] += num/15728640;
        num%=15728640;
        nums[i] += num/983040*16;
        num%=983040;
        nums[i] += num/65536;
        num%=65536;
        nums[i] += num/4096*16;
        num%=4096;
        nums[i] += num/256;
        num%=256;
        nums[i] += num/16*16;
        num%=16;
        nums[i] += num/1;
        num%=1;

        nums[i] %= b;
        if(nums[i] < c)
            map.put(nums[i], map.getOrDefault(nums[i], 0) + 1);
    }
    int res = 0;
    for(int val : map.values()){
        res = Math.max(val, res);
    }
    System.out.println(res);
}
}
```