Двоично тегло

„Двоично Тегло” на едно число е броят единици, които то има в двоичния си запис. Например теглото на 42 е 3, тъй като двоичното му представяне е 101010. 1337 от друга страна има тегло 6. Вашата задача е да напишете програма, която бързо смята тази сума. Генерирането на числата става по следния начин: в началото се определят пет цели числа N, F, A, B, и M. Редицата има N члена, първият от които е остатъка на F при деление на M. Всеки следващ се образува, като предходният се умножи по A, после се прибави B и се вземе остатъка на резултата при деление на M. Тоест: 1. S1 = F % M 2. Si = (Si-1 * A + B) % M, за i = 2…N. Гореописаната процедура се нарича Linear Congruential Generator (LCG) и е един от най-старите и разпространени начини за генериране на псевдослучайни числа.

Жокер: Ако пробваме по наивния начин (да генерираме всички числа и на всяко да му отделим битовете) бихме извършили максимално 10,000,000 * 30 (всяко число е максимум 30 бита) = 3 * 10^9, което няма да мине по време.
Идея: Добре е да използуваме таблица, в която предварително да изчислим броя на битовете за всички числа от 1 до 1,000,000,007. След това като генерираме текущото число от редицата само с една операция (вместо 30) ще получим броят на битовете му като прочетем елемента от таблицата с индекс това число.
Проблем 1: Таблицата ще съдържа 1,000,000,007 ~= 10^9 елемента (един милиард), а максимално разрешената ни памет в състезателното програмиране е 512MB = char масив с 512 милиона елемента или 256MB = char масив с 256 милиона елемента в зависимост от системата за автоматично оценяване на задачите. Тук в hackerrank ограничението по памет е 512MB, но РСОП може да се проведе на друга платформа с ограничение 256MB. Проблем 2: Времето за пресмятане на таблицата би било 1,000,000,007 * 30 (всяко число е максимум 30 бита) = 30 * 10^9, което е 10 пъти по бавно от наивното решение. Решение: Да направим таблицата много много по-малка и за всяко число да гледаме два пъти в нея и да сумираме резултатите от тези две гледания. Вие решете колко по-малка трябва да е тя и на кои два индекса в таблицата трябва да гледаме за всяко число за да вземем броят на битовете на две части от числото, които конкатинирани образуват цялото число. Примерно двоичното число 100111011 можем да разбием на 1001 | 11011 и от предварително генерираната таблица директно да прочетем броят на битовете на 1001 = 2 и броят на битовете на 11011 = 4. Сигурни вече се досещате колко е броят на битовете на 100111011, като знаем че този на 1001 = 2

и този на 11011 = 4. Генерацията на таблица се прави само веднъж в началото на програмата и се ползува за всеки тест.

Input Format

На първия ред на стандартния вход ще бъде зададен броят тестове T. Всеки от тестовете ще съдържа по един ред с петте цели числа N, F, A, B, M.

Constraints

1 ≤ T ≤ 40 1 ≤ N ≤ 10,000,000 1 ≤ F, A, B, M ≤ 1,000,000,007 Броят числа, сумарно за всички тестове, да е по-малък от 100,000,000.

Output Format

За всеки тест изведете по един ред с едно единствено число – сумата на теглата на генерираните числа.

Sample Input 0

```
3
5 42 3 5 113
1337 7 17 9 12345
10000000 1337 666 42 1000000000
```

Sample Output 0

```
17
8837
133468731
```