

Heaven_ 与砖头

time limit: 2000ms

Heaven_ 要去搬砖。

现在他面前有 n 个砖头从左到右依次排列，第 i 块砖头的重量为 w_i ($0 \leq w_i < 2147483647$)，他每次只能从最左边或者最右边取走一块砖头，每搬一块砖头将会使他增加 $v \times w$ 点疲劳值，其中 v 是他的疲劳值，且初始疲劳值为 1。

他发现面前这 n 个砖头的重量虽然表面上杂乱无章，但是背后其实有一个固定的模式在支配着每一块砖头的重量。他很快发现了这些砖头的重量排列的规律：这些砖头的重量竟然仅仅与两个固定的正整数 $k1, k2$ 有关！这种规律最终可以简化成为下面的 C++ 代码所描述的形式：

```
unsigned long long k1, k2;
unsigned long long WeightOfNextBrick(){
    ull k3 = k1, k4 = k2;
    k1 = k4;
    k3 ^= k3 << 23;
    k2 = k3 * k4 ^ (k3 >> 17) ^ (k4 >> 26);
    return (k2 + k4) % 2147483647;
}
```

解释：当 $k1, k2$ 确定以后，每次运行函数 `WeightOfNextBrick()` 的返回值即为下一块砖头的重量。i.e. 第一次执行得到的返回值为第一块砖头的重量，第二次的返回值为第二块砖头得重量，...，第 n 次的返回值为第 n 块砖头的重量。

Heaven_ 搬完所有的砖后必须去相亲一次，而相亲将使他的疲劳值降低一点，但他希望尽可能的增加他的疲劳值。所以请你告诉他，他最终能达到的最大疲劳值是多少。

Input

第一行为一个正整数 T ($T \leq 5$)，表示测试样例的数目。

每个测试样例包含两行，第一行包含一个正整数 n ($1 \leq n \leq 10^8$)，表示砖头的个数。

第二行包含两个正整数 $k1, k2$ ($0 \leq k1, k2 \leq 100000$)，表示两个能够确定全部砖头重量的两个正整数（具体含义见题目描述）

Output

对于每个测试样例，输出一行，包含一个整数，表示 Heaven_ 所能达到的最大疲劳值。

由于他可能会超级无敌疲劳，你只需要将他的最大疲劳值对 $10^9 + 7$ 取模输出即可。

Sample Input

```
1
2
8 9
```

Sample Output

```
33094569
```