Problem G. Glad You Came

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Steve has an integer array a of length n (1-based). He assigned all the elements as zero at the beginning. After that, he made m operations, each of which is to update an interval of a with some value. You need to figure out $\bigoplus_{i=1}^{n} (i \cdot a_i)$ after all his operations are finished, where $\bigoplus$ means the bitwise exclusive-OR operator.

In order to avoid huge input data, these operations are encrypted through some particular approach.

There are three unsigned 32-bit integers X, Y and Z which have initial values given by the input. A random number generator function is described as following, where $\wedge$ means the bitwise exclusive-OR operator, $<<$ means the bitwise left shift operator and $>>$ means the bitwise right shift operator. Note that function would change the values of X, Y and Z after calling.

```
function RNG61()
    X ← X ∧ (X << 11)   ▷ 32-bit unsigned integer overflow might occur
    X ← X ∧ (X >> 4)
    X ← X ∧ (X << 5)    ▷ 32-bit unsigned integer overflow might occur
    X ← X ∧ (X >> 14)
    W ← X ∧ (Y ∧ Z)     ▷ as a partial 32-bit unsigned integer
    X ← Y
    Y ← Z
    Z ← W
    return Z
end function
```

Let the i-th result value of calling the above function as f_i $(i = 1, 2, \cdots, 3m)$. The i-th operation of Steve is to update a_j as v_i if $a_j < v_i$ $(j = l_i, l_i + 1, \cdots, r_i)$, where

$$\begin{cases} l_i &= \min\left((f_{3i-2} \bmod n) + 1, (f_{3i-1} \bmod n) + 1\right) \\ r_i &= \max\left((f_{3i-2} \bmod n) + 1, (f_{3i-1} \bmod n) + 1\right) \; (i = 1, 2, \cdots, m). \\ v_i &= f_{3i} \bmod 2^{30} \end{cases}$$

Input

The first line contains one integer T, indicating the number of test cases.

Each of the following T lines describes a test case and contains five space-separated integers n, m, X, Y and Z.

$1 \le T \le 100$, $1 \le n \le 10^5$, $1 \le m \le 5 \cdot 10^6$, $0 \le X, Y, Z < 2^{30}$.

It is guaranteed that the sum of n in all the test cases does not exceed 10^6 and the sum of m in all the test cases does not exceed $5 \cdot 10^7$.

Output

For each test case, output the answer in one line.

Example

standard input	standard output
4 1 10 100 1000 10000 10 100 1000 10000 100000 100 1000 10000 100000 1000000 1000 10000 100000 1000000 10000000	1031463378 1446334207 351511856 47320301347

Note

In the first sample, $a = [1031463378]$ after all the operations.

In the second sample, $a = [1036205629, 1064909195, 1044643689, 1062944339, 1062944339, 1062944339, 1062944339, 1057472915, 1057472915, 1030626924]$ after all the operations.