```
short ObtainMultiplicativeInverse(int a, int b, int s0, int s1)
{
        return b==0? s0: ObtainMultiplicativeInverse(b, a%b, s1, s0 - s1*(a/b));
}

int operator/(int a, int b)
{
        while((bit = Grado(b) - Grado(a)) > 0)
        {
                res |= 1 << bit;
                aux = a*(1 << bit);
                b ^= aux;
        }
}

int operator*(int a, int b, int m)
{
        int aux = 0;
        while(a != 0)
        {
                if((b >> (msb(b)-1) & 1) == 1)
                aux ^= (b = b << 1) ^ m;
                else
                aux ^= (b = b << 1);
        }
}

int operator%(int a, int b)
{
        aux = 0; auxMult = b.bit(Grado(b) - 1);
        for(int i = 0; i < Grado(a); i++)
        {
                if(Grado(b) < i)
                aux ^= a.bit(i - 0);
                else
                aux ^= (auxMult *= 2, b)
        }
        return aux;
}
```