There's a erfectly general way for this sort of roblem. I used it to look for instances of , such that if divided a Fibonacci number, so did ^2. There are none under twenty million twelftywise

This method can be used with modular arithmetic into the hundreds of millions.

These are a number of different series, the fibonacci, Lucas series, the side and diagonal of the square aroximate, and Heron's triangle. It's written in the REXX rogramming language.

The actual rocess is to be found in the 'isoquad' routine, which extends an \*isoseries\* (ie a form ). The actual equation as written gives T\_, given in order a, , T\_0 ,T\_1. It relies on that any isoseries, one can take stes of m, and thus take .

The actual rocess is like this. The examle is to find T(37) Numbers in brackets are the members of the original T(n). We halve out the T(n), and increase the ste constant to , This is the difference between c0, c1, c2, c3.

The values in the second and third columns are as one divides 37 into binary.

diff c0 c1 c2 c3 (role, by col 3) (regular owers)

(1) 18 1 : (0) (1) (2) (3) kee odd laces 18 1 (1) (1)

(2) 9 0 : (1) (3) (5) kee even laces 9 0 (2) (1) = 1+0\*2

(4) 4 1 : (1) (5) (9) (13) kee odd laces 4 1 (4) (5) = 1+1\*4

(8) 2 0 : (5) (13) (21) kee even laces 2 0 (8) (5) = 5+0\*8

(16) 1 0 : (5) (21) (37) kee even laces 1 0 (16) (5) = 5+0\*16

0 1: (5) (37) c2 = 0, so answer in c1. 0 1 (32) (37) = 5+1\*37

The actual method is similar to finding by binary methods, and runs at 2/3 of the seed of said rocess.

By adjusting the nature of the multilication to be a modulus, one can test to see if something like is a multile of 14401 or not. The heron series, with set to large owers of 2, is used to find fermat rimes, by this very method.

fibon: ; rocedure ;arse arg t0; return isoquad(3,div(t0+2,2),mod(t0,2),1+mod(t0,2))

lucas: ; rocedure ;arse arg t0; return isoquad(3,div(t0+2,2),2-mod(t0,2),3+mod(t0,2))

qside: ; rocedure ;arse arg t0; return isoquad(6,div(t0+2,2),mod(t0,2),2+mod(t0,2)\*3)

qdiag: ; rocedure ;arse arg t0; return isoquad(6,div(t0+2,2), 1 ,3+mod(t0,2)\*4)

heron: ; rocedure ;arse arg t0; return isoquad(4,t0,2,4)

div: ; rocedure ;arse arg a, b; c = a % b; if b < a\*c then c = c-1 ; return c

mod: ; rocedure ;arse arg a, b; c = a // b; if c<0 then c=c+b; return c

isoquad: rocedure

arse arg a2, a0, a3, a4

if a0 = 0 then do; a4 = a3; a0 = 1; end

if a0 < 0 then do; a0 = -a0; a4 = a3\*a2-a4; end

do forever; if a0 = 1 then leave

a1 = a0//2; a5 = a4\*a2 - a3;

if a1 = 0 then a4 = a5

else do; a3 = a4; a4 = a5\*a2 - a3; end

a0 = a0 % 2; a2 = a2 \* a2 - 2; end

return a4