Triangle Coding

Our system has found a way to use a variable x to plays the role of two variables (x_a, x_b). To achieve this, three different programs are invented. The first function f_{enc} produces on unique natural number $\frac{x_a \times (x_a+1)}{2} + x_b$. To achieve this, two programs are invented f_a and f_b. As x increases starting from 0, the value of x can be decoded into a pair $(f_b(x) - f_a(x), f_b(x))$ taking the following values: $(0,0),(1,0),(1,1),(2,0),(2,1),(2,2),(3,0),\ldots$. The Python's representation of these functions is:

```
def fa(X):
  x = X
  for y in range (1,(2 + (X // 5)) + 1):
    x = x - (y if (y - x) <= 0 else 0)
  return x
def fb(X):
  x = X
  for y in range (1,(2 + (X // (1 + (2 + 2)))) + 1):
    x = x - (0 if x <= 0 else (1 + y))
  return x
```

The function `f10(x)` is calculating $b - a$, and the function `f11(x)` is calculating b where the input x is enumerated by pair (a, b).

This representation was initially discovered when our system found solutions for the sequences A2262 [1], and A25581 [2]. Indeed, f_a is a solution for A2262 invented in the first generation and $-f_b$ is a solution for A25581 invented during the 9^{th} generation.

The designed language for our programs only supports a maximum of two variables. The triangle coding essentially allows to unpack two variables $(f_b(x) - f_a(x), f_b(x))$ from one x. To pack Therefore, the triangle coding turns out to be very useful in calculating more complex programs that require more than two variables. One example of such program where triangle coding is used in this way is A279364 [3] sum of 5th powers of proper divisors of n. One can easily implement it as:

```
def f(X):
    res = 0
    for y in range(1,X+1):
        res = res + (y**5 if (X+1) % y <= 0 else 0)
    return res
```

Note however that there are three variables used in the body of the loop, in particular `res`, `y`, and `X`. Since the native language supports only two variables available at each moment, this Python code cannot be straightforwardly translated into the language of our programs. Nevertheless, the program generator has found a workaround using the triangle coding – it packs `X` and `y` into a single natural number, so it can perform the summing loop adding to `res`, and in order to decode what to add to `res`, it unpacks the pair, and checks whether `y` divides `X+1`.

[1] https://oeis.org/A002262

[2] https://oeis.org/A025581

[3] https://oeis.org/A279364