The result for wensely is 0 = 0, 0 = 0, 0 = 0

```
Needs["RISC`fastZeil`"];
需要
Needs["RISC`Dependencies`"];
需要
(*Load Aligator*)
<< Aligator`

Aligator[WHILE[(d ≥ e), IF[(P < a + b), b := b / 2;
    d := d / 2, a := a + b;
    y := y + d / 2;
    b := b / 2;
    d := d / 2]]]
```

Fast Zeilberger Package version 3.61
written by Peter Paule, Markus Schorn, and Axel Riese
Copyright Research Institute for Symbolic Computation (RISC),
Johannes Kepler University, Linz, Austria

Package FindRelations version 0.29
written by Manuel Kauers and Burkhard Zimmermann
Copyright Research Institute for Symbolic Computation (RISC),
Johannes Kepler University, Linz, Austria

Package Dependencies version 0.30
written by Manuel Kauers and Burkhard Zimmermann
Copyright Research Institute for Symbolic Computation (RISC),
Johannes Kepler University, Linz, Austria

```
Aligator.m
Automated Loop Invariant Generation by Algebraic Techniques Over the Rationals.
Package written by Laura Kovacs and Andreas Humenberger — © TU Wien — V 0.6 (2017-02-03)
```

```
Out[4]= {-b[5] × d[0] + b[0] × d[5], a[0] × d[5] - a[5] × d[5] + 2 b[5] (-y[0] + y[5]),
 a[0] × d[0] - a[5] × d[0] + 2 b[0] (-y[0] + y[5])}
```

(this means bd = bd, ad = ad + 2 b (y – y), ad = ad + 2 b (–y + y))

Our environment is correctly configured

In[5]:=
```
Aligator[WHILE[y > 0, t1 := t2;
  t2 := a;
  a := 5 (n + 2) t2 + 6 (n^2 + 3 n + 2) t1;
  b := 2 b;
  c := 3 (n + 2) c;
  d := (n + 2) d], LoopCounter → n, IniVal → {t1 := 1;
   t2 := 1;
   a := 1;
   b := 1;
   c := 1;
   d := 1}]
```

```
Method is complete!
```

Out[5]= $\{(7 a - 138 b c)^2 - 256 d^2, a - 23 b c + t2, 49 a^2 - 1104 a b c - 256 d^2 + 828 b c\, t2, -256 d^2 + (a - 6 t2)^2, -a + 6 t1 + 5 t2\}$