

```

/* CalcAzero solves for T, P, Q when  $W = \pm 1 \pmod T$ ,  $A = 0$ ,  $B = 1$  *
* in the eqn:  $N(TPPB/TQP B + \text{sign} * B) = N(Z/V)$  */

```

```

void CalcAzero (int Z, int v, int A, int sign, int B) {
    int T, P, MAX, Q;
    MAX = abs (Z - (sign*A));
    for (T = 1; T < MAX + 1; T++) {
        for (P = 1; P < MAX + 1; P++) {
            if (T * P * P * B == Z) {
                if ((v - sign * B) % (T * P * B) == 0) {
                    Q = (v - sign * B) / (T * P * B);
                    SolveU (Z, v, A, sign, B, T, P, Q);
                }
            }
        }
    }
}

```

```

/* Calc solves for T, P, Q when  $W = \pm 1 \pmod T$  in the eqn:
 $TP[P B - A Q] + \text{sign} * A = Z$  when A not zero */

```

```

void Calc (int Z, int V, int A, int sign, int B) {
    int SUM, Q, MAX, P, T;
    SUM = -99;
    Q = -99;
    MAX = abs (Z - (sign * A));

    for (T = 1; T < MAX + 1; T++) {
        for (P = 1; P < MAX + 1; P++) {
            if ((Z - (sign*A))%(T*P) == 0) {
                SUM = (Z - (sign * A)) / (T * P);          /* SUM = pb- aq*/
                if ((SUM - P * B) % A == 0) {
                    Q = (SUM - P * B) / (-A);
                    SolveU (Z, V, A, sign, B, T, P, Q);
                }
            }
        }
    }
}

```

```

void solveunor3 (int A, int B, int Z, int V) {
    int sign, v, b;

```

```

// NOTE: f1 and g1 are ignored at the moment
/* input N(A/B) and N(Z/V). */

```

```

b = BinversemodA (B, A);
if ( b < -1 ) {
    sprintf (comment, "A (%d) and B (%d) must be relatively prime", A,
B);

```

*← checks  $\text{gcd}(A, B) = 1$*

$kp (ps - q) + a$   
 $kp (ps - q) - a$

```

error_exit (comment);
}
v = BinversemodA (V, Z);
if ( v < -1 ){
    sprintf (comment, "V (%d) and Z (%d) must be relatively prime", V,
Z);
    error_exit (comment);
}
Calc (Z, V, A, B); // Calc solves for T when W not \pm 1 mod T
sign = 1; // sign refers to the \pm sign
// in +IP[PB - AQ] \pm A| = Z
if (A != 0) {
    Calc (Z, V, A, sign, B); // Calc solves for T when W = \pm 1 mod T
    Calc (-Z, -V, A, sign, B);
    sign = -1;
    Calc (Z, V, A, sign, B);
    Calc (-Z, -V, A, sign, B);
}
sign = 1;
if (A == 0) {
    if (B < -1 || B == 0 || B > 1)
        error_exit ("A and B must be relatively prime. N(A, B) = N(0, it is
1)");
    if (Z < 0) {
        Z = -Z;
        V = -V;
    }
    CalcAzero (Z, V, A, sign, 1);
    if (v != V)
        CalcAzero (Z, v, A, sign, 1);
    sign = -1;
    CalcAzero (Z, V, A, sign, 1);
    if (v != V)
        CalcAzero (Z, v, A, sign, 1);
}
}
}

```

$gcd(V, Z) = 1$   
 $N\left(\frac{Z}{V}\right) = N\left(\frac{Z}{\tilde{V}}\right)$  where  $\tilde{V} \equiv 1 \pmod{Z}$   
 $N\left(\frac{Z}{V+kZ}\right)$   $N\left(\frac{Z}{\tilde{V}+kZ}\right)$

$N\left(\frac{Z}{V}\right), N\left(\frac{-Z}{-V}\right)$

**NOTE**  $\tilde{V}$  case missing here but handled later in program

```

/* Note the following is not the complete main */
int main (int argc, char **argv) {
    int A, B, Z, V;
    solveunor3 (A, B, Z, V);
}

```

start 5

$A = 29$   
 $B = 6$   
 $Z = 1$   
 $V = 0$