International Journal of Scientific Engineering and Science
Volume 8, Issue 2, pp. 1-6, 2024.
ISSN (Online): 2456-7361

Significance of Continued Fraction to Solve Binary Quadratic Equations

V. Pandichelvi[1], R. Vanaja[2]

[1]Assistant Professor, PG & Research Department of Mathematics, Urumu Dhanalakshmi College, Trichy *(Affiliated to Bharathidasan University)*

[2]Assistant Professor, PG & Research Department of Mathematics, Shrimati Indira Gandhi College, Trichy *(Affiliated to Bharathidasan University)*

E-mail: [1]mvpmahesh2017@gmail.com, [2]vanajvicky09@gmail.com

Abstract*—In this paper, the procedure for devising solutions of distinct categories of the Diophantine equations* ${x_n}^2 - (p^2q^2 + Np){y_n}^2 = (N^2)^n$, $N \in Z - \{0\}$ *in terms of generalised Fibonacci and Lucas sequences where* p *and* q *are cousin primes, twin primes and safe primes are deliberated by retaining the concept of continued fraction.*

Keywords*— Binary Quadratic Equations, Continued Fraction, Fibonacci sequence, Lucas sequence.*

I. INTRODUCTION

The Fibonacci sequence, familiarized by Leonardo de Pisa [4] in 1202, is specified by the recurrence relation $f_{n+2} = f_{n+1} + f_n, n \geq 0$ with $f_0 = 0, f_1 = 1$. The Lucas sequence is named after Edouard Lucas and has the similar recurrence relation but different initial conditions $l_{n+2} = l_{n+1} + l_n, n \geq 0$ with $l_0 = 2$ and $l_0 = 1$. Generalised Fibonacci and generalised Lucas sequences are defined for two non-zero numbers s and t satisfying $s^2 + 4t > 0$, $F_{n+2}(s,t) = sF_{n+1}(s,t) + tF_n(s,t), n \geq 0$ with $F_0(s,t) = 0, F_1(s,t) = 1$ and $L_{n+2}(s,t) = sL_{n+1}(s,t) + tL_n(s,t)$ with $L_0(s,t) = 2, L_1(s,t) = s$. Binet's formulae for these two demarcated sequences are $F_n(s,t) = \frac{\alpha^n - \beta^n}{\alpha - \beta}$ and $L_n(s,t) = \alpha^n + \beta^n$ where $\alpha = \frac{s + \sqrt{s^2+4t}}{2}$, $\beta = \frac{s + \sqrt{s^2+4t}}{2}$ are the roots of the equation $x^2 - sx - t = 0$ such that $\alpha + \beta = s$ and $\alpha - \beta = \sqrt{s^2 + 4t}$, $\alpha\beta = -t$.

A Pell's equation is a Diophantine equation of the form $x^2 - Dy^2 = 1$ where $x, y \in Z$ and D is a square free natural number. If $\sqrt{D}$ is square free, then the development of continued fraction of $\sqrt{D}$ is stated by $\sqrt{D} = [a_0 : \overline{a_1, a_2, \ldots, a_{n-1}, a_n}]$, $a_n = 2a_0$ where n is the length of period $\alpha_0 = \sqrt{D}, \alpha_k = \lfloor \alpha_k \rfloor, \alpha_{k+1} = \frac{1}{\alpha_k - a_k}, k = 0,1,2,3\ etc$, $\frac{p_i}{q_i} = [a_0 : \overline{a_1, a_2, \ldots, a_{i-1}, a_i}]$ is i[th] convergent of $\sqrt{D}$ for $i \geq 0$. Numerous authors have used several tactics to solve various Diophantine equations in [1-9]. In this article, the application of continued fraction to treasure the solutions of the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ where $N \in Z - \{0\}, D = p^2q^2 + Np$, p and q are cousin primes, twin primes and safe primes are inspected.

II. DETECTION OF INTEGER SOLUTIONS TO PELL'S TYPE EQUATION BY MEANS OF CONTINUED FRACTION

The possible solutions of the Diophantine equation ${x_n}^2 - (p^2q^2 + Np)\,{y_n}^2 = (N^2)^n$ where $N \in Z - \{0\}$, p, q are cousin primes, twin primes and safe primes are analysed in the following theorems.

Theorem 2.1

If p, q are cousin primes and $D = p^2q^2 + Np = p^4 + 8p^3 + 16p^2 + Np$ which is not a perfect square, then

(i) $\sqrt{D} = \left[p^2 + 4p : \overline{\frac{2(p+4)}{N}, 2(p^2+4p)}\right]$

(ii) The smallest solution entailing cousin primes to the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ is
$(x_1, y_1) = (2p^3 + 16p^2 + 32p + N, 2p + 8)$

(iii) All other solutions (x_n, y_n) for a natural number $n \geq 1$ to the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ in terms of generalised Fibonacci and Lucas sequences linking cousin primes are enumerated by

$x_n = \frac{1}{2} L_n(4p^3 + 32p^2 + 64p + 2N, -N^2)$

$y_n = \frac{2p+8}{2} F_n(4p^3 + 32p^2 + 64p + 2N, -N^2)$

Proof

(i) $\sqrt{D} = (p^2 + 4p) + \sqrt{p^4 + 8p^3 + 16p^2 + Np} - (p^2 + 4p)$

$= (p^2 + 4p) + \frac{1}{\frac{2(p+4)}{N} + \frac{\sqrt{p^4+8p^3+16p^2+p} - (p^2+4p)}{Np}}$

$= (p^2 + 4p) + \frac{1}{\frac{2(p+4)}{N} + \frac{1}{2(p^2+4p) + \sqrt{p^4+8p^3+16p^2+p} - (p^2+4p)}}$

$= \left[p^2 + 4p : \overline{\frac{2(p+4)}{N}, 2(p^2+4p)}\right]$

(ii) Note that the length of the period is an even number 2. Then, the smallest solution to the desired equation is provided by the relation

$\frac{p_1}{q_1} = (p^2 + 4p) + \frac{1}{\frac{2(p+4)}{N}} = \frac{2p^3 + 16p^2 + 32p + N}{2(p+4)}$

So, the smallest solution of the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ involving cousin primes is
$(x_1, y_1) = (2p^3 + 16p^2 + 32p + N, 2p + 8)$

(iii) All other solutions (x_n, y_n), $n \geq 1$ connecting cousin primes are listed by

$x_n + y_n\sqrt{D} = \left(2p^3 + 16p^2 + 32p + N + 2(p+4)\sqrt{D}\right)^n$

$x_n - y_n\sqrt{D} = \left(2p^3 + 16p^2 + 32p + N - 2(p+4)\sqrt{D}\right)^n$

Choose the transformations as

$\alpha = 2p^3 + 16p^2 + 32p + N + 2(p+4)\sqrt{D}$

Volume 8, Issue 2, pp. 1-6, 2024.
ISSN (Online): 2456-7361

$$\beta = 2p^3 + 16p^2 + 32p + N - 2(p+4)\sqrt{D}$$

Then, the solutions to the equation in terms of generalised Fibonacci and Lucas sequences linking cousin primes are monitored by

$$x_n = \frac{\alpha^n + \beta^n}{2} = \frac{1}{2} L_n(4p^3 + 32p^2 + 64p + 2N, -N^2)$$

$$y_n = \frac{\alpha^n - \beta^n}{2\sqrt{D}} = (2p+8) F_n(4p^3 + 32p^2 + 64p + 2N, -1)$$

Arithmetical illustrations of theorem 2.1 is presented in the following table.

p	N	n	L_n	F_n	x_n	y_n	${x_n}^2 - D{y_n}^2$	$(N^2)^n$
3	-2	1	584	1	292	14	4	4
		2	341048	584	170524	8176	16	16
		3	19,91,69,696	341052	99584848	4774728	64	64
	-1	1	586	1	293	14	1	1
		2	343394	586	171697	8204	1	1
		3	201228298	343395	100614149	4807530	1	1
	1	1	590	1	295	14	1	1
		2	348098	590	174049	8260	1	1
		3	205377230	348099	102688615	4873386	1	1
	2	1	592	1	296	14	4	4
		2	350456	592	175228	8288	16	16
		3	20,74,67,584	350460	10,37,33,792	4906440	64	64
7	-2	1	3384	1	1692	22	4	4
		2	11451448	3384	5725724	74448	16	16
		3	38751686496	11451452	19375843248	251931944	64	64
	-1	1	3386	1	1693	22	1	1
		2	11464994	3386	5732497	74492	1	1
		3	38820466298	11464995	19410233149	252229890	1	1
	1	1	3390	1	1695	22.00	1	1
		2	11492098	3390	5746049	74580.00	1	1
		3	38958208830	11492099	19479104415	252826178	1	1
	2	1	3392	1	1696	22	4	4
		2	11505656	3392	5752828	74624	16	16
		3	39027171584	11505660	19513585792	253124520	64	64

All plausible solutions to the equation in the previous theorem can be verified by the succeeding Python program 1.

Python program 1

```
from decimal import Decimal, getcontext
# Set the precision for Decimal calculations
getcontext().prec
= 40 # You can adjust the precision as needed
def lucas(n, s, t):
  if n < 0:
    return "Incorrect input"
  elif n == 0:
    return 2
  elif n == 1:
    return s
  else:
    a, b = 2, s
    for _ in range(2, n + 1):
      a, b = b, s * b + t * a
    return b
def fibonacci(n, s, t):
  if n < 0:
    return "Incorrect input"
  elif n == 0:
    return 0
  elif n == 1:
    return 1
  else:
    a, b = 0, 1
    for _ in range(2, n + 1):
      a, b = b, s * b + t * a
    return b
def calculate_values(N, p, n):
  D = (p * p * p * p) + (8 * p * p * p) + (16 * p
        * p) + (N * p)
  s = 2 * (2 * p * p * p + 16 * p * p + 32 * p
        + N)
  t = -N ** 2
  x_n = lucas(n, s, t) / 2
  x2 = x_n ** 2
  y_n = (2 * p + 8) * fibonacci(n, s, t)
  y2 = y_n ** 2
  D_y2 = D * y2
  h = x2 - D_y2
  return D, x_n, x2, y_n, y2, D_y2, abs(h)
# Get user inputs
N = int(input('Enter The Value Of N: '))
p = int(input('Enter The Value Of p: '))
n = int(input('Enter The Value Of n: '))
D, x_n, x2, y_n, y2, D_y2, h
        = calculate_values(N, p, n)
# Print the results with proper formatting using Decimal
print(f'D = {D}')
print(f'x(n) = {x_n}')
print(f'x(n)^2 = {x2}')
```

Volume 8, Issue 2, pp. 1-6, 2024. ISSN (Online): 2456-7361

```
    print(f'y(n) = {y_n}')
    print(f'y(n)^2 = {y2}')
    print(f'D * y(n)^2 = {D_y2}')
    print(f'h = {h}')
    if(h == N ** (2 * n)):
  print("x and y are the solutions of the given equation")
    else:
    print("x and y are not the solutions of the given equation")
```

Output

```
    Enter The Value Of N: -2
    Enter The Value Of p: 3
    Enter The Value Of n: 3
    D = 435
    x(n) = 99584848
    x(n)^2 = 9917141951183104
    y(n) = 4774728
    y(n)^2 = 22798027473984
    D * y(n)^2 = 9917141951183040
    h = 64
x and y are the solutions of the given equation
    Enter The Value Of N: 15
    Enter The Value Of p: 7
    Enter The Value Of n: 2
    D = 6034
    x(n) = 5841137
    x(n)^2 = 34118881452769
    y(n) = 75196
    y(n)^2 = 5654438416
    D * y(n)^2 = 34118881402144
    h = 50625
x and y are the solutions of the given equation
    Enter The Value Of N: 4
    Enter The Value Of p: 11
    Enter The Value Of n: 2
    D = 27269
    x(n) = 49084216
    x(n)^2 = 2409260260334656
    y(n) = 297240
    y(n)^2 = 88351617600
    D * y(n)^2 = 2409260260334400
    h = 256
x and y are the solutions of the given equation
```

Theorem 2.2

If p, q are twin primes, $D = p^2q^2 + Np = p^4 + 4p^3 + 4p^2 + Np$ such that D is not a perfect square, then

(i) $\sqrt{D} = \left[p^2 + 2p : \overline{\frac{2(p+2)}{N}, 2(p^2 + 2p)}\right]$

(ii) The least solution concerning twin primes to the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ is $(x_1, y_1) = (2p^3 + 8p^2 + 8p + 1, 2p + 4)$

(iii) All possible solutions (x_n, y_n) for a natural number $n \geq 1$ in generalised Fibonacci and Lucas sequences relating twin primes to the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ are prearranged by

$$x_n = \frac{1}{2}L_n(4p^3 + 16p^2 + 16p + 2, -N^2)$$
$$y_n = (2p + 4)F_n(4p^3 + 16p^2 + 16p + 2N, -N^2)$$

Proof

The proof of theorem 2.2 is analogous to theorem 2.1

Algebraic calculations of solutions to theorem 2.2 is mentioned in the table below.

p	N	n	L_n	F_n	x_n	y_n	${x_n}^2 - D{y_n}^2$	$(N^2)^n$
3	-2	1	296	1	148	10	4	4
		2	87608	296	43804	2960	16	16
		3	25930784	87612	12965392	876120	64	64
	-1	1	586	1	293	14	1	1
		2	343394	586	171697	8204	1	1
		3	201228298	343395	100614149	4807530	1	1
	1	1	302	1	151	10	1	1
		2	91202	302	45601	3020	1	1
		3	27542702	91203	13771351	912030	1	1
	2	1	304	1	152	10	4	4
		2	92408	304	46204	3040	16	16
		3	28090816	92412	14045408	924120	64	64
5	-2	1	976	1	488	14	4	4
		2	952568	976	476284	13664	16	16
		3	929702464	952572	464851232	13336008	64	64
	-1	1	3386	1	1693	22	1	1
		2	11464994	3386	5732497	74492	1	1
		3	38820466298	11464995	19410233149	252229890	1	1
	1	1	2270	1	1135	18	1	1
		2	5152898	2270	2576449	40860	1	1
		3	11697076190	5152899	5848538095	92752182	1	1
	2	1	984	1	492	14	4	4
		2	968248	984	484124	13776	16	16
		3	952752096	968252	476376048	13555528	64	64

Entire solutions to the equation in the preceding theorem can be substantiated by the subsequent Python program 2.

Python Program 2

```
from decimal import Decimal, getcontext
# Set the precision for Decimal calculations
```

```
getcontext().prec
= 40 # You can adjust the precision as needed
def lucas(n, s, t):
  if n < 0:
    return "Incorrect input"
  elif n == 0:
    return 2
  elif n == 1:
    return s
  else:
    a, b = 2, s
    for _ in range(2, n + 1):
      a, b = b, s * b + t * a
    return b
def fibonacci(n, s, t):
  if n < 0:
    return "Incorrect input"
  elif n == 0:
    return 0
  elif n == 1:
    return 1
  else:
    a, b = 0, 1
    for _ in range(2, n + 1):
      a, b = b, s * b + t * a
    return b
def calculate_values(N, p, n):
  D = (p * p * p * p) + (4 * p * p * p) + (4 * p
                 * p) + (N * p)
  s = 2 * (2 * p * p * p + 8 * p * p + 8 * p + N)
  t = -N ** 2
  x_n = lucas(n, s, t) / 2
  x2 = x_n ** 2
  y_n = (2 * p + 4) * fibonacci(n, s, t)
  y2 = y_n ** 2
  D_y2 = D * y2
  h = x2 - D_y2
  return D, x_n, x2, y_n, y2, D_y2, abs(h)
# Get user inputs
N = int(input('Enter The Value Of N: '))
p = int(input('Enter The Value Of p: '))
n = int(input('Enter The Value Of n: '))
D, x_n, x2, y_n, y2, D_y2, h
                 = calculate_values(N, p, n)
# Print the results with proper formatting using Decimal
print(f'D = {D}')
print(f'x(n) = {x_n}')
print(f'x(n)^2 = {x2}')
print(f'y(n) = {y_n}')
print(f'y(n)^2 = {y2}')
print(f'D * y(n)^2 = {D_y2}')
print(f'h = {h}')
if(h == N ** (2 * n)):
  print("x and y are the solutions of the given equation")
else:
 print("x and y are not the solutions of the given equation")
```

Output

```
Enter The Value Of N: 2
Enter The Value Of p: 3
Enter The Value Of n: 2
D = 231
x(n) = 46204
x(n)^2 = 2134809616
y(n) = 3040
y(n)^2 = 9241600
D * y(n)^2 = 2134809600
h = 16
x and y are the solutions of the given equation
Enter The Value Of N: -2
Enter The Value Of p: 5
Enter The Value Of n: 3
D = 1215
x(n) = 464851232
x(n)^2 = 216086667891917824
y(n) = 13336008
y(n)^2 = 177849109376064
D * y(n)^2 = 216086667891917760
h = 64
x and y are the solutions of the given equation
```

Theorem 2.3

If $D = p^2q^2 + Np = 4p^4 + 4p^3 + p^2 + Np$ where p, q are safe primes and D is not a square number, then

(i) $\sqrt{D} = \left[2p^2 + p; \overline{\frac{2(2p+1)}{N}, 2(2p^2+p)}\right]$

(ii) The lowest solution in safe primes to the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ is
$(x_1, y_1) = (8p^3 + 8p^2 + 2p + 1, 4p + 2)$

(iii) All possible solutions (x_n, y_n) for a natural number $n \geq 1$ in generalised Fibonacci and Lucas sequences involving safe primes to the equation ${x_n}^2 - D{y_n}^2 = (N^2)^n$ are given by

$$x_n = \frac{1}{2} L_n(16p^3 + 16p^2 + 4p + 2N, -N^2)$$
$$y_n = (4p + 2)F_n(16p^3 + 16p^2 + 4p + 2N, -N^2)$$

Proof

The proof of theorem 2.3 is similar to theorem 2.1

Numerical calculation of solutions presented in theorem 2.3 is tabulated below.

p	N	n	L_n	F_n	x_n	y_n	${x_n}^2 - D{y_n}^2$	$(N^2)^n$
3	-2	1	584	1	292	14	4	4
		2	341048	584	170524	8176	16	16
		3	199169696	341052	99584848	4774728	64	64
	-1	1	586	1	293	14	1	1
		2	343394	586	171697	8204	1	1
		3	201228298	343395	100614149	4807530	1	1

International Journal of Scientific Engineering and Science
Volume 8, Issue 2, pp. 1-6, 2024. ISSN (Online): 2456-7361

	1	1	590	1	295	14	1	1
		2	348098	590	174049	8260	1	1
		3	205377230	348099	102688615	4873386	1	1
	2	1	584	1	292	14	4	4
		2	341048	584	170524	8176	16	16
		3	199169696	341052	99584848	4774728	64	64
5	-2	1	2416	1	1208	22	4	4
		2	5837048	2416	2918524	53152	16	16
		3	14102298304	5837052	7051149152	128415144	64	64
	-1	1	2418	1	1209	22	1	1
		2	5846722	2418	2923361	53196	1	1
		3	14137371378	5846723	7068685689	128627906	1	1
	1	1	2422	1	1211	22	1	1
		2	5866082	2422	2933041	53284	1	1
		3	14207648182	5866083	7103824091	129053826	1	1
	2	1	2424	1	1212	22	4	4
		2	5875768	2424	2937884	53328	16	16
		3	14242851936	5875772	7121425968	129266984	64	64

The complete solutions to the equation in the earlier theorem can be validated by the ensuing Python program 3.

Python Program 3

```
from decimal import Decimal, getcontext
# Set the precision for Decimal calculations
getcontext().prec
= 40 # You can adjust the precision as needed
def lucas(n, s, t):
  if n < 0:
    return "Incorrect input"
  elif n == 0:
    return 2
  elif n == 1:
    return s
  else:
    a, b = 2, s
    for _ in range(2, n + 1):
      a, b = b, s * b + t * a
    return b
def fibonacci(n, s, t):
  if n < 0:
    return "Incorrect input"
  elif n == 0:
    return 0
  elif n == 1:
    return 1
  else:
    a, b = 0, 1
    for _ in range(2, n + 1):
      a, b = b, s * b + t * a
    return b
def calculate_values(N, p, n):
  D = (4 * p * p * p * p) + (4 * p * p * p) + (p
          * p) + (N * p)
  s = 2 * (8 * p * p * p + 8 * p * p + 2 * p + N)
  t = -N ** 2
  x_n = lucas(n, s, t) / 2
  x2 = x_n ** 2
  y_n = (4 * p + 2) * fibonacci(n, s, t)
  y2 = y_n ** 2
  D_y2 = D * y2
  h = x2 - D_y2
  return D, x_n, x2, y_n, y2, D_y2, abs(h)
# Get user inputs
N = int(input('Enter The Value Of N: '))
p = int(input('Enter The Value Of p: '))
n = int(input('Enter The Value Of n: '))
D, x_n, x2, y_n, y2, D_y2, h
        = calculate_values(N, p, n)
# Print the results with proper formatting using Decimal
print(f'D = {D}')
print(f'x(n) = {x_n}')
print(f'x(n)^2 = {x2}')
print(f'y(n) = {y_n}')
print(f'y(n)^2 = {y2}')
print(f'D * y(n)^2 = {D_y2}')
print(f'h = {h}')
if(h == N ** (2 * n)):
  print("x and y are the solutions of the given equation")
else:
  print("x and y are not the solutions of the given equation")
```

Output

```
Enter The Value Of N: − 2
Enter The Value Of p: 3
Enter The Value Of n: 3
D = 435
x(n) = 99584848
x(n)^2 = 9917141951183104
y(n) = 4774728
y(n)^2 = 22798027473984
D * y(n)^2 = 9917141951183040
h = 64
x and y are the solutions of the given equation
Enter The Value Of N: 2
Enter The Value Of p: 5
Enter The Value Of n: 1
D = 3035
x(n) = 1212
x(n)^2 = 1468944
y(n) = 22
```

$y(n)^2 = 484$
$D * y(n)^2 = 1468940$
$h = 4$
x and y are the solutions of the given equation

III. CONCLUSION

This article analyses the solutions of the equations ${x_n}^2 - (p^2q^2 + Np){y_n}^2 = (N^2)^n, \quad N \in Z - \{0\}$ using generalised Fibonacci and generalised Lucas sequences where p and q are cousin primes, twin primes and safe primes. Similarly, one can look for integer solutions to identical types of Diophantine equations in terms of generalised Pell and generalised Pell Lucas sequences.

REFERENCES

1. Guney, Merve. "Solutions of the Pell equations $x^2 - (a^2b^2 + 2b)y^2 = N$ when $N \in \{1, 4\}$," *Mathematica Aeterna*, 2,7, 629-638, 2012.
2. Bilge Peker, Hasan Senay, "Solutions of the Parametric Pell Equation $x^2 - (b^2 - b)y^2 = N$," via Generalized Fibonacci and Lucas Numbers", *Int. J. Contemp. Math. Sciences*, 8, 13, 637 – 642, 2013.
3. Jafari-Petroudia, Seyyed Hossein, and Behzad Pirouzb. "On some properties of (k, h) −Pell sequence and (k, h) −Pell-Lucas ssequence," *Int. J. Adv. Appl. Math. and Mech*, 3, 1, 98-101, 2015.
4. Koshy, Thomas, "Fibonacci and Lucas Numbers with Applications", *John Wiley & Sons*, 2, 2019.
5. R. Keskin, M.G. Duman, "Positive integer solutions of some Pell equations," *Palest. J. Math*, 8, 2, 213–226, 2019.
6. Roji Bala, Vinod Mishra, "Examples and Counterexamples," *Elsevier B.V*, 2, 1-4, 2022.
7. V. Pandichelvi, S. Saranya, "Implication of Continued Fraction and Metallic Ratios to Resolve Binary Quadratic Diophantine Equations," *International Journal of Mathematics Trends and Technology*, 68, 12, 13-15, 2022.
8. V. Pandichelvi P. Sandhya, "Assessment of Solutions in Pell and Pell-Lucas numbers to Disparate Polynomial Equations of Degree Two", *Wesleyan Journal of Research*, 14, 5, 129-134, 2021.
9. V. Pandichelvi P. Sandhya, "Conception of Positive Integer Solutions Relating Jacobsthal and Jacobsthal – Lucas Numbers to Restricted Number of Quadratic Equations with Double Variables", *Journal of Data Acquisition and Processing*, 38, 1, 2201-2207, 2023.ss