http://www.TheoryOfEverything.org

Even FibBinary Numbers and the Golden Ratio

J Gregory Moxness[*]
TheoryOfEverything.org

(Dated: January 1, 2019)

Previously, a determination of the relationship between the Natural numbers (N) and the n'th odd fibbinary number has been made using a relationship with the Golden ratio $\phi = (\sqrt{5}+1)/2$ and $\tau = 1/\phi$. Specifically, if the n'th odd fibbinary equates to the j'th N, then $j = \lfloor n\phi^2 \rfloor - 1$. This note documents the completion of the relationship for the even fibbinary numbers, such that if the n'th even fibbinary equates to the j'th N, then $j = \lfloor n\phi \rfloor - 1$, starting at $j = 0$ for $n = 1$. Alternatively, starting at $j = 2$ for $n = 1$, then $j = \lfloor n\phi + \tau \rfloor$.

PACS numbers: 02.10.Ox
Keywords: Combinatorics

I. INTRODUCTION

For an introduction to the topic, please reference [1], [2] and [3].

II. VERIFICATON FOR ODD AND EVEN FIBBINARY NUMBERS

Fig. 1 shows the related elements of the odd fibbinaries and Fig. 2 shows the related elements of the even fibbinaries. These involve the conversions between the decimal, binary, and Gray codes, along with the numeric and symbolic *MacMahongraph* compositions. The tables list row elements for $j = N \leq 98$ for odd $n \leq 38$ and $j = N \leq 100$ for even $n \leq 63$.

It is interesting to note that like the approximation of the ratio of two consecutive Fibonacci numbers F_n/F_{n-1} to the Golden ratio, the ratio of the length of the even vs. odd list of n's will approximate ϕ as $n \to \infty$ as well, as it should given the ratio of ϕ in the structure of their respective functions.

The *Mathematica*^{TM} code used to create the tables below also confirms the relationships and patterns from [1]. Fig. 3 shows some example outputs for these functions. Fig. 4 show some code snippets used in this analysis.

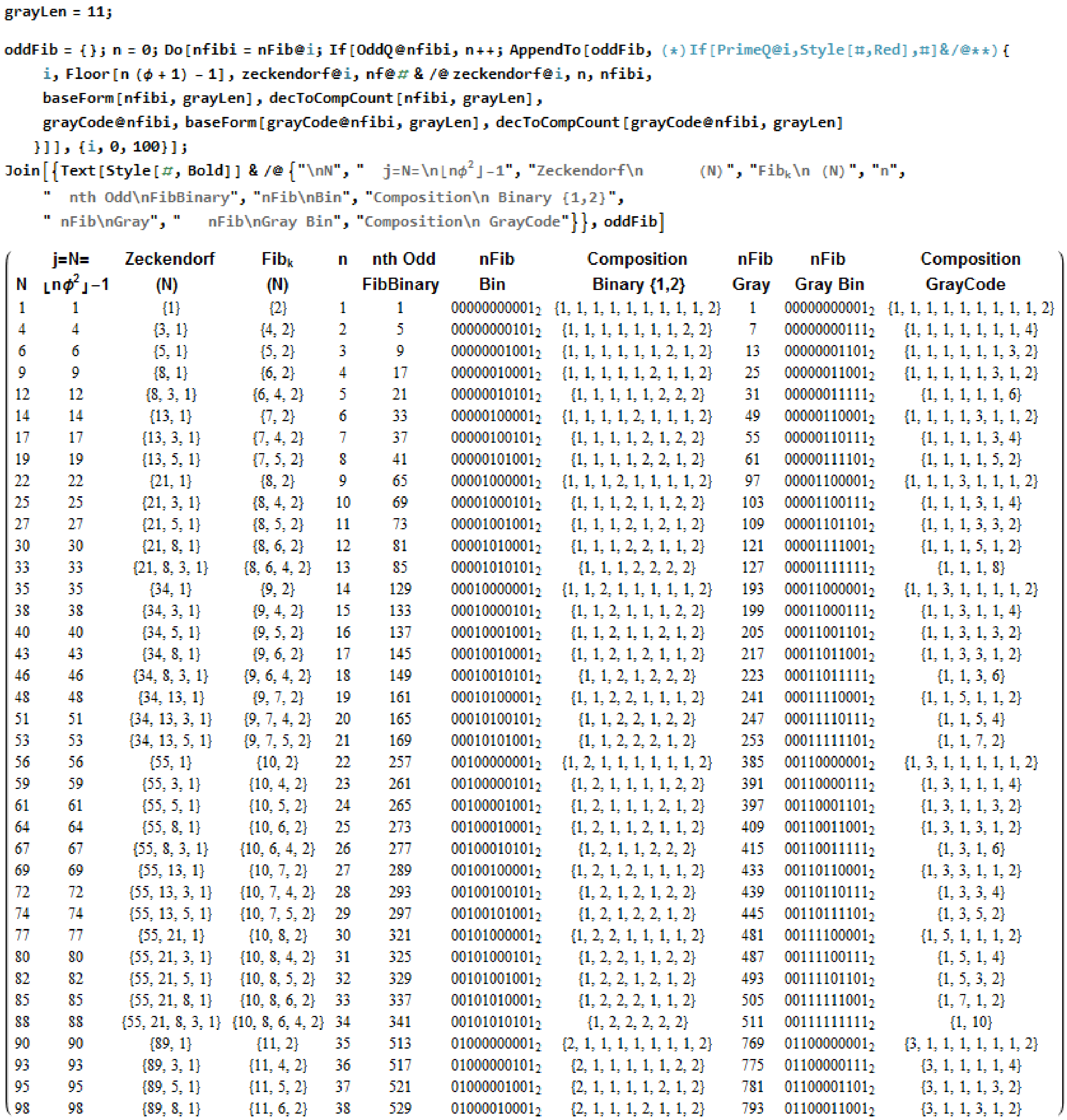
III. CONCLUSION

While the proof of the even fibbinary numbers sequence is not yet formulated in this quick note, the symmetry of the pattern compared to the odd fibbinary numbers combined with the confirmation for n to several million is reassuring.

Acknowledgments

I would like to thank my wife for her love and patience and those in academia who have taken the time to review this work.

[*] URL: http://www.TheoryOfEverything.org/TOE/JGM; mailto:jgmoxness@TheoryOfEverything.org

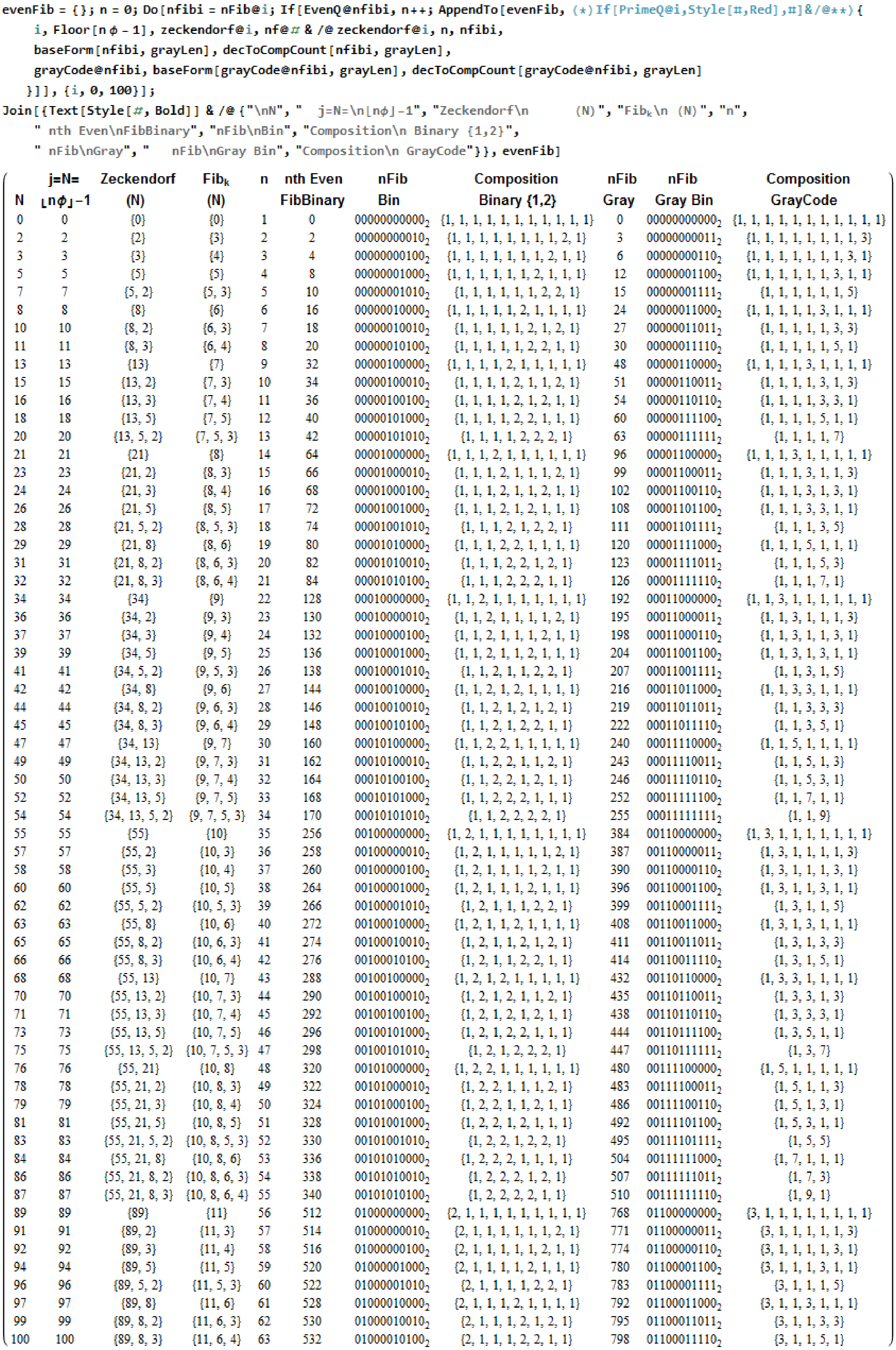


```
grayLen = 11;

oddFib = {}; n = 0; Do[nfibi = nFib@i; If[OddQ@nfibi, n++; AppendTo[oddFib, (*)If[PrimeQ@i,Style[#,Red],#]&/@**){
    i, Floor[n (ϕ + 1) - 1], zeckendorf@i, nf@# & /@ zeckendorf@i, n, nfibi,
    baseForm[nfibi, grayLen], decToCompCount[nfibi, grayLen],
    grayCode@nfibi, baseForm[grayCode@nfibi, grayLen], decToCompCount[grayCode@nfibi, grayLen]
   }]], {i, 0, 100}];
Join[{Text[Style[#, Bold]] & /@ {"\nN", "  j=N=\n⌊nϕ²⌋-1", "Zeckendorf\n       (N)", "Fibₖ\n (N)", "n",
    "  nth Odd\nFibBinary", "nFib\nBin", "Composition\n Binary {1,2}",
    " nFib\nGray", "   nFib\nGray Bin", "Composition\n GrayCode"}}, oddFib]
```

	j=N=	Zeckendorf	Fib$_k$	n	nth Odd	nFib	Composition	nFib	nFib	Composition
N	$\lfloor n\phi^2 \rfloor-1$	(N)	(N)		FibBinary	Bin	Binary {1,2}	Gray	Gray Bin	GrayCode
1	1	{1}	{2}	1	1	00000000001_2	{1, 1, 1, 1, 1, 1, 1, 1, 1, 2}	1	00000000001_2	{1, 1, 1, 1, 1, 1, 1, 1, 1, 2}
4	4	{3, 1}	{4, 2}	2	5	00000000101_2	{1, 1, 1, 1, 1, 1, 1, 2, 2}	7	00000000111_2	{1, 1, 1, 1, 1, 1, 1, 4}
6	6	{5, 1}	{5, 2}	3	9	00000001001_2	{1, 1, 1, 1, 1, 1, 2, 1, 2}	13	00000001101_2	{1, 1, 1, 1, 1, 1, 3, 2}
9	9	{8, 1}	{6, 2}	4	17	00000010001_2	{1, 1, 1, 1, 1, 2, 1, 1, 2}	25	00000011001_2	{1, 1, 1, 1, 1, 3, 1, 2}
12	12	{8, 3, 1}	{6, 4, 2}	5	21	00000010101_2	{1, 1, 1, 1, 1, 2, 2, 2}	31	00000011111_2	{1, 1, 1, 1, 1, 6}
14	14	{13, 1}	{7, 2}	6	33	00000100001_2	{1, 1, 1, 1, 2, 1, 1, 1, 2}	49	00000110001_2	{1, 1, 1, 1, 3, 1, 1, 2}
17	17	{13, 3, 1}	{7, 4, 2}	7	37	00000100101_2	{1, 1, 1, 1, 2, 1, 2, 2}	55	00000110111_2	{1, 1, 1, 1, 3, 4}
19	19	{13, 5, 1}	{7, 5, 2}	8	41	00000101001_2	{1, 1, 1, 1, 2, 2, 1, 2}	61	00000111101_2	{1, 1, 1, 1, 5, 2}
22	22	{21, 1}	{8, 2}	9	65	00001000001_2	{1, 1, 1, 2, 1, 1, 1, 1, 2}	97	00001100001_2	{1, 1, 1, 3, 1, 1, 1, 2}
25	25	{21, 3, 1}	{8, 4, 2}	10	69	00001000101_2	{1, 1, 1, 2, 1, 1, 2, 2}	103	00001100111_2	{1, 1, 1, 3, 1, 4}
27	27	{21, 5, 1}	{8, 5, 2}	11	73	00001001001_2	{1, 1, 1, 2, 1, 2, 1, 2}	109	00001101101_2	{1, 1, 1, 3, 3, 2}
30	30	{21, 8, 1}	{8, 6, 2}	12	81	00001010001_2	{1, 1, 1, 2, 2, 1, 1, 2}	121	00001111001_2	{1, 1, 1, 5, 1, 2}
33	33	{21, 8, 3, 1}	{8, 6, 4, 2}	13	85	00001010101_2	{1, 1, 1, 2, 2, 2, 2}	127	00001111111_2	{1, 1, 1, 8}
35	35	{34, 1}	{9, 2}	14	129	00010000001_2	{1, 1, 2, 1, 1, 1, 1, 1, 2}	193	00011000001_2	{1, 1, 3, 1, 1, 1, 1, 2}
38	38	{34, 3, 1}	{9, 4, 2}	15	133	00010000101_2	{1, 1, 2, 1, 1, 1, 2, 2}	199	00011000111_2	{1, 1, 3, 1, 1, 4}
40	40	{34, 5, 1}	{9, 5, 2}	16	137	00010001001_2	{1, 1, 2, 1, 1, 2, 1, 2}	205	00011001101_2	{1, 1, 3, 1, 3, 2}
43	43	{34, 8, 1}	{9, 6, 2}	17	145	00010010001_2	{1, 1, 2, 1, 2, 1, 1, 2}	217	00011011001_2	{1, 1, 3, 3, 1, 2}
46	46	{34, 8, 3, 1}	{9, 6, 4, 2}	18	149	00010010101_2	{1, 1, 2, 1, 2, 2, 2}	223	00011011111_2	{1, 1, 3, 6}
48	48	{34, 13, 1}	{9, 7, 2}	19	161	00010100001_2	{1, 1, 2, 2, 1, 1, 1, 2}	241	00011110001_2	{1, 1, 5, 1, 1, 2}
51	51	{34, 13, 3, 1}	{9, 7, 4, 2}	20	165	00010100101_2	{1, 1, 2, 2, 1, 2, 2}	247	00011110111_2	{1, 1, 5, 4}
53	53	{34, 13, 5, 1}	{9, 7, 5, 2}	21	169	00010101001_2	{1, 1, 2, 2, 2, 1, 2}	253	00011111101_2	{1, 1, 7, 2}
56	56	{55, 1}	{10, 2}	22	257	00100000001_2	{1, 2, 1, 1, 1, 1, 1, 1, 2}	385	00110000001_2	{1, 3, 1, 1, 1, 1, 1, 2}
59	59	{55, 3, 1}	{10, 4, 2}	23	261	00100000101_2	{1, 2, 1, 1, 1, 1, 2, 2}	391	00110000111_2	{1, 3, 1, 1, 1, 4}
61	61	{55, 5, 1}	{10, 5, 2}	24	265	00100001001_2	{1, 2, 1, 1, 1, 2, 1, 2}	397	00110001101_2	{1, 3, 1, 1, 3, 2}
64	64	{55, 8, 1}	{10, 6, 2}	25	273	00100010001_2	{1, 2, 1, 1, 2, 1, 1, 2}	409	00110011001_2	{1, 3, 1, 3, 1, 2}
67	67	{55, 8, 3, 1}	{10, 6, 4, 2}	26	277	00100010101_2	{1, 2, 1, 1, 2, 2, 2}	415	00110011111_2	{1, 3, 1, 6}
69	69	{55, 13, 1}	{10, 7, 2}	27	289	00100100001_2	{1, 2, 1, 2, 1, 1, 1, 2}	433	00110110001_2	{1, 3, 3, 1, 1, 2}
72	72	{55, 13, 3, 1}	{10, 7, 4, 2}	28	293	00100100101_2	{1, 2, 1, 2, 1, 2, 2}	439	00110110111_2	{1, 3, 3, 4}
74	74	{55, 13, 5, 1}	{10, 7, 5, 2}	29	297	00100101001_2	{1, 2, 1, 2, 2, 1, 2}	445	00110111101_2	{1, 3, 5, 2}
77	77	{55, 21, 1}	{10, 8, 2}	30	321	00101000001_2	{1, 2, 2, 1, 1, 1, 1, 2}	481	00111100001_2	{1, 5, 1, 1, 1, 2}
80	80	{55, 21, 3, 1}	{10, 8, 4, 2}	31	325	00101000101_2	{1, 2, 2, 1, 1, 2, 2}	487	00111100111_2	{1, 5, 1, 4}
82	82	{55, 21, 5, 1}	{10, 8, 5, 2}	32	329	00101001001_2	{1, 2, 2, 1, 2, 1, 2}	493	00111101101_2	{1, 5, 3, 2}
85	85	{55, 21, 8, 1}	{10, 8, 6, 2}	33	337	00101010001_2	{1, 2, 2, 2, 1, 1, 2}	505	00111111001_2	{1, 7, 1, 2}
88	88	{55, 21, 8, 3, 1}	{10, 8, 6, 4, 2}	34	341	00101010101_2	{1, 2, 2, 2, 2, 2}	511	00111111111_2	{1, 10}
90	90	{89, 1}	{11, 2}	35	513	01000000001_2	{2, 1, 1, 1, 1, 1, 1, 1, 2}	769	01100000001_2	{3, 1, 1, 1, 1, 1, 1, 2}
93	93	{89, 3, 1}	{11, 4, 2}	36	517	01000000101_2	{2, 1, 1, 1, 1, 1, 2, 2}	775	01100000111_2	{3, 1, 1, 1, 1, 4}
95	95	{89, 5, 1}	{11, 5, 2}	37	521	01000001001_2	{2, 1, 1, 1, 1, 2, 1, 2}	781	01100001101_2	{3, 1, 1, 1, 3, 2}
98	98	{89, 8, 1}	{11, 6, 2}	38	529	01000010001_2	{2, 1, 1, 1, 2, 1, 1, 2}	793	01100011001_2	{3, 1, 1, 3, 1, 2}

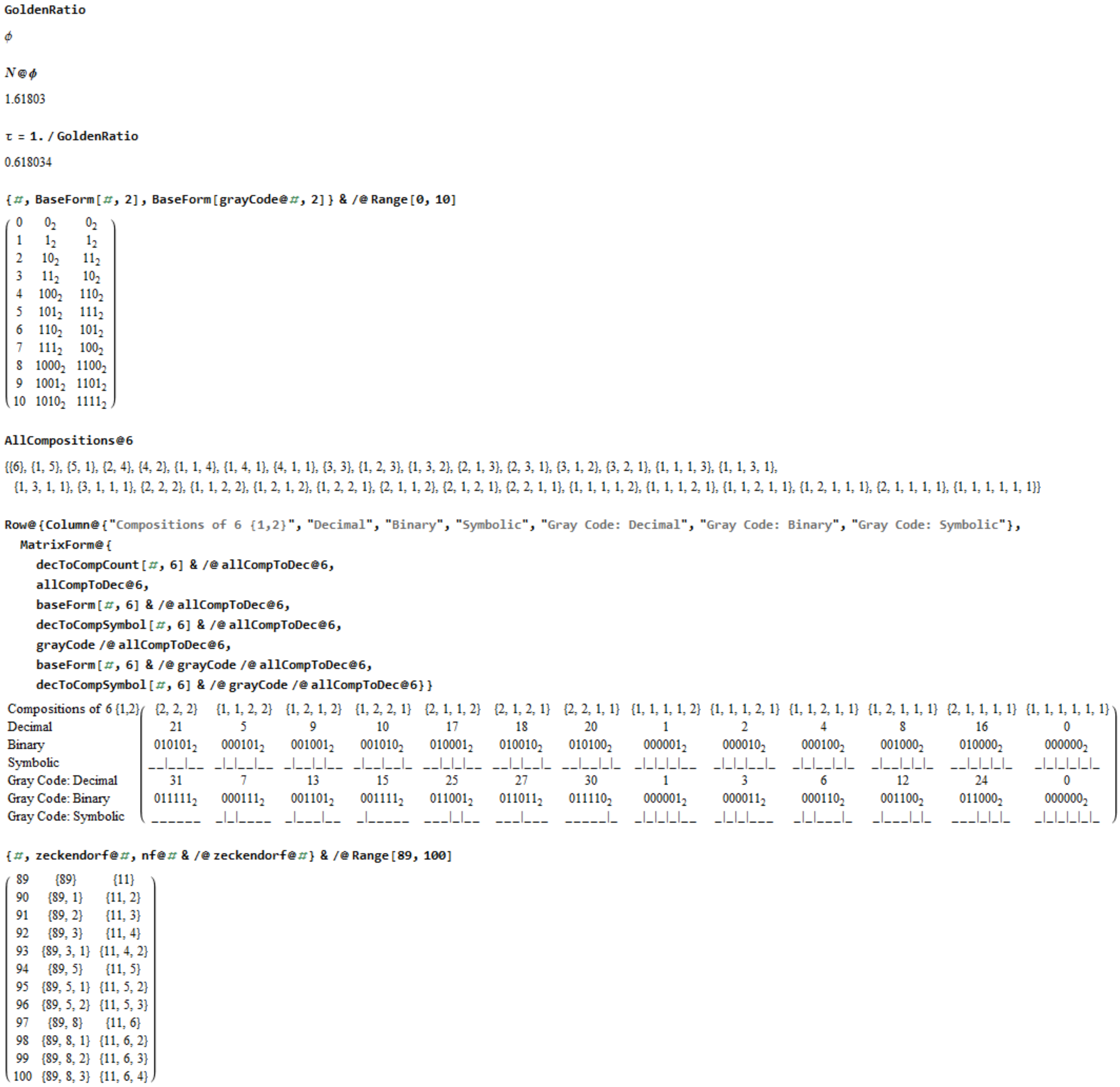
FIG. 1: Comprehensive list of odd fibbinary related elements for $j = N \leq 98$ for odd $n \leq 38$



```
evenFib = {}; n = 0; Do[nfibi = nFib@i; If[EvenQ@nfibi, n++; AppendTo[evenFib, (*)If[PrimeQ@i,Style[#,Red],#]&/@**){
    i, Floor[n ϕ - 1], zeckendorf@i, nf@# & /@ zeckendorf@i, n, nfibi,
    baseForm[nfibi, grayLen], decToCompCount[nfibi, grayLen],
    grayCode@nfibi, baseForm[grayCode@nfibi, grayLen], decToCompCount[grayCode@nfibi, grayLen]
   }]], {i, 0, 100}];
Join[{Text[Style[#, Bold]] & /@ {"\nN", "  j=N=\n⌊nϕ⌋-1", "Zeckendorf\n      (N)", "Fibₖ\n (N)", "n",
    " nth Even\nFibBinary", "nFib\nBin", "Composition\n Binary {1,2}",
    " nFib\nGray", "   nFib\nGray Bin", "Composition\n GrayCode"}}, evenFib]
```

	j=N=	Zeckendorf	Fib$_k$	n	nth Even	nFib	Composition	nFib	nFib	Composition
N	⌊nϕ⌋−1	(N)	(N)		FibBinary	Bin	Binary {1,2}	Gray	Gray Bin	GrayCode
0	0	{0}	{0}	1	0	00000000000_2	{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}	0	00000000000_2	{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
2	2	{2}	{3}	2	2	00000000010_2	{1, 1, 1, 1, 1, 1, 1, 1, 2, 1}	3	00000000011_2	{1, 1, 1, 1, 1, 1, 1, 1, 3}
3	3	{3}	{4}	3	4	00000000100_2	{1, 1, 1, 1, 1, 1, 1, 2, 1, 1}	6	00000000110_2	{1, 1, 1, 1, 1, 1, 1, 3, 1}
5	5	{5}	{5}	4	8	00000001000_2	{1, 1, 1, 1, 1, 1, 2, 1, 1, 1}	12	00000001100_2	{1, 1, 1, 1, 1, 1, 3, 1, 1}
7	7	{5, 2}	{5, 3}	5	10	00000001010_2	{1, 1, 1, 1, 1, 1, 2, 2, 1}	15	00000001111_2	{1, 1, 1, 1, 1, 1, 5}
8	8	{8}	{6}	6	16	00000010000_2	{1, 1, 1, 1, 1, 2, 1, 1, 1, 1}	24	00000011000_2	{1, 1, 1, 1, 1, 3, 1, 1, 1}
10	10	{8, 2}	{6, 3}	7	18	00000010010_2	{1, 1, 1, 1, 1, 2, 1, 2, 1}	27	00000011011_2	{1, 1, 1, 1, 1, 3, 3}
11	11	{8, 3}	{6, 4}	8	20	00000010100_2	{1, 1, 1, 1, 1, 2, 2, 1, 1}	30	00000011110_2	{1, 1, 1, 1, 1, 5, 1}
13	13	{13}	{7}	9	32	00000100000_2	{1, 1, 1, 1, 2, 1, 1, 1, 1, 1}	48	00000110000_2	{1, 1, 1, 1, 3, 1, 1, 1, 1}
15	15	{13, 2}	{7, 3}	10	34	00000100010_2	{1, 1, 1, 1, 2, 1, 1, 2, 1}	51	00000110011_2	{1, 1, 1, 1, 3, 1, 3}
16	16	{13, 3}	{7, 4}	11	36	00000100100_2	{1, 1, 1, 1, 2, 1, 2, 1, 1}	54	00000110110_2	{1, 1, 1, 1, 3, 3, 1}
18	18	{13, 5}	{7, 5}	12	40	00000101000_2	{1, 1, 1, 1, 2, 2, 1, 1, 1}	60	00000111100_2	{1, 1, 1, 1, 5, 1, 1}
20	20	{13, 5, 2}	{7, 5, 3}	13	42	00000101010_2	{1, 1, 1, 1, 2, 2, 2, 1}	63	00000111111_2	{1, 1, 1, 1, 7}
21	21	{21}	{8}	14	64	00001000000_2	{1, 1, 1, 2, 1, 1, 1, 1, 1, 1}	96	00001100000_2	{1, 1, 1, 3, 1, 1, 1, 1, 1}
23	23	{21, 2}	{8, 3}	15	66	00001000010_2	{1, 1, 1, 2, 1, 1, 1, 2, 1}	99	00001100011_2	{1, 1, 1, 3, 1, 1, 3}
24	24	{21, 3}	{8, 4}	16	68	00001000100_2	{1, 1, 1, 2, 1, 1, 2, 1, 1}	102	00001100110_2	{1, 1, 1, 3, 1, 3, 1}
26	26	{21, 5}	{8, 5}	17	72	00001001000_2	{1, 1, 1, 2, 1, 2, 1, 1, 1}	108	00001101100_2	{1, 1, 1, 3, 3, 1, 1}
28	28	{21, 5, 2}	{8, 5, 3}	18	74	00001001010_2	{1, 1, 1, 2, 1, 2, 2, 1}	111	00001101111_2	{1, 1, 1, 3, 5}
29	29	{21, 8}	{8, 6}	19	80	00001010000_2	{1, 1, 1, 2, 2, 1, 1, 1, 1}	120	00001111000_2	{1, 1, 1, 5, 1, 1, 1}
31	31	{21, 8, 2}	{8, 6, 3}	20	82	00001010010_2	{1, 1, 1, 2, 2, 1, 2, 1}	123	00001111011_2	{1, 1, 1, 5, 3}
32	32	{21, 8, 3}	{8, 6, 4}	21	84	00001010100_2	{1, 1, 1, 2, 2, 2, 1, 1}	126	00001111110_2	{1, 1, 1, 7, 1}
34	34	{34}	{9}	22	128	00010000000_2	{1, 1, 2, 1, 1, 1, 1, 1, 1, 1}	192	00011000000_2	{1, 1, 3, 1, 1, 1, 1, 1, 1}
36	36	{34, 2}	{9, 3}	23	130	00010000010_2	{1, 1, 2, 1, 1, 1, 1, 2, 1}	195	00011000011_2	{1, 1, 3, 1, 1, 1, 3}
37	37	{34, 3}	{9, 4}	24	132	00010000100_2	{1, 1, 2, 1, 1, 1, 2, 1, 1}	198	00011000110_2	{1, 1, 3, 1, 1, 3, 1}
39	39	{34, 5}	{9, 5}	25	136	00010001000_2	{1, 1, 2, 1, 1, 2, 1, 1, 1}	204	00011001100_2	{1, 1, 3, 1, 3, 1, 1}
41	41	{34, 5, 2}	{9, 5, 3}	26	138	00010001010_2	{1, 1, 2, 1, 1, 2, 2, 1}	207	00011001111_2	{1, 1, 3, 1, 5}
42	42	{34, 8}	{9, 6}	27	144	00010010000_2	{1, 1, 2, 1, 2, 1, 1, 1, 1}	216	00011011000_2	{1, 1, 3, 3, 1, 1, 1}
44	44	{34, 8, 2}	{9, 6, 3}	28	146	00010010010_2	{1, 1, 2, 1, 2, 1, 2, 1}	219	00011011011_2	{1, 1, 3, 3, 3}
45	45	{34, 8, 3}	{9, 6, 4}	29	148	00010010100_2	{1, 1, 2, 1, 2, 2, 1, 1}	222	00011011110_2	{1, 1, 3, 5, 1}
47	47	{34, 13}	{9, 7}	30	160	00010100000_2	{1, 1, 2, 2, 1, 1, 1, 1, 1}	240	00011110000_2	{1, 1, 5, 1, 1, 1, 1}
49	49	{34, 13, 2}	{9, 7, 3}	31	162	00010100010_2	{1, 1, 2, 2, 1, 1, 2, 1}	243	00011110011_2	{1, 1, 5, 1, 3}
50	50	{34, 13, 3}	{9, 7, 4}	32	164	00010100100_2	{1, 1, 2, 2, 1, 2, 1, 1}	246	00011110110_2	{1, 1, 5, 3, 1}
52	52	{34, 13, 5}	{9, 7, 5}	33	168	00010101000_2	{1, 1, 2, 2, 2, 1, 1, 1}	252	00011111100_2	{1, 1, 7, 1, 1}
54	54	{34, 13, 5, 2}	{9, 7, 5, 3}	34	170	00010101010_2	{1, 1, 2, 2, 2, 2, 1}	255	00011111111_2	{1, 1, 9}
55	55	{55}	{10}	35	256	00100000000_2	{1, 2, 1, 1, 1, 1, 1, 1, 1, 1}	384	00110000000_2	{1, 3, 1, 1, 1, 1, 1, 1, 1}
57	57	{55, 2}	{10, 3}	36	258	00100000010_2	{1, 2, 1, 1, 1, 1, 1, 2, 1}	387	00110000011_2	{1, 3, 1, 1, 1, 1, 3}
58	58	{55, 3}	{10, 4}	37	260	00100000100_2	{1, 2, 1, 1, 1, 1, 2, 1, 1}	390	00110000110_2	{1, 3, 1, 1, 1, 3, 1}
60	60	{55, 5}	{10, 5}	38	264	00100001000_2	{1, 2, 1, 1, 1, 2, 1, 1, 1}	396	00110001100_2	{1, 3, 1, 1, 3, 1, 1}
62	62	{55, 5, 2}	{10, 5, 3}	39	266	00100001010_2	{1, 2, 1, 1, 1, 2, 2, 1}	399	00110001111_2	{1, 3, 1, 1, 5}
63	63	{55, 8}	{10, 6}	40	272	00100010000_2	{1, 2, 1, 1, 2, 1, 1, 1, 1}	408	00110011000_2	{1, 3, 1, 3, 1, 1, 1}
65	65	{55, 8, 2}	{10, 6, 3}	41	274	00100010010_2	{1, 2, 1, 1, 2, 1, 2, 1}	411	00110011011_2	{1, 3, 1, 3, 3}
66	66	{55, 8, 3}	{10, 6, 4}	42	276	00100010100_2	{1, 2, 1, 1, 2, 2, 1, 1}	414	00110011110_2	{1, 3, 1, 5, 1}
68	68	{55, 13}	{10, 7}	43	288	00100100000_2	{1, 2, 1, 2, 1, 1, 1, 1, 1}	432	00110110000_2	{1, 3, 3, 1, 1, 1, 1}
70	70	{55, 13, 2}	{10, 7, 3}	44	290	00100100010_2	{1, 2, 1, 2, 1, 1, 2, 1}	435	00110110011_2	{1, 3, 3, 1, 3}
71	71	{55, 13, 3}	{10, 7, 4}	45	292	00100100100_2	{1, 2, 1, 2, 1, 2, 1, 1}	438	00110110110_2	{1, 3, 3, 3, 1}
73	73	{55, 13, 5}	{10, 7, 5}	46	296	00100101000_2	{1, 2, 1, 2, 2, 1, 1, 1}	444	00110111100_2	{1, 3, 5, 1, 1}
75	75	{55, 13, 5, 2}	{10, 7, 5, 3}	47	298	00100101010_2	{1, 2, 1, 2, 2, 2, 1}	447	00110111111_2	{1, 3, 7}
76	76	{55, 21}	{10, 8}	48	320	00101000000_2	{1, 2, 2, 1, 1, 1, 1, 1, 1}	480	00111100000_2	{1, 5, 1, 1, 1, 1, 1}
78	78	{55, 21, 2}	{10, 8, 3}	49	322	00101000010_2	{1, 2, 2, 1, 1, 1, 2, 1}	483	00111100011_2	{1, 5, 1, 1, 3}
79	79	{55, 21, 3}	{10, 8, 4}	50	324	00101000100_2	{1, 2, 2, 1, 1, 2, 1, 1}	486	00111100110_2	{1, 5, 1, 3, 1}
81	81	{55, 21, 5}	{10, 8, 5}	51	328	00101001000_2	{1, 2, 2, 1, 2, 1, 1, 1}	492	00111101100_2	{1, 5, 3, 1, 1}
83	83	{55, 21, 5, 2}	{10, 8, 5, 3}	52	330	00101001010_2	{1, 2, 2, 1, 2, 2, 1}	495	00111101111_2	{1, 5, 5}
84	84	{55, 21, 8}	{10, 8, 6}	53	336	00101010000_2	{1, 2, 2, 2, 1, 1, 1, 1}	504	00111111000_2	{1, 7, 1, 1, 1}
86	86	{55, 21, 8, 2}	{10, 8, 6, 3}	54	338	00101010010_2	{1, 2, 2, 2, 1, 2, 1}	507	00111111011_2	{1, 7, 3}
87	87	{55, 21, 8, 3}	{10, 8, 6, 4}	55	340	00101010100_2	{1, 2, 2, 2, 2, 1, 1}	510	00111111110_2	{1, 9, 1}
89	89	{89}	{11}	56	512	01000000000_2	{2, 1, 1, 1, 1, 1, 1, 1, 1, 1}	768	01100000000_2	{3, 1, 1, 1, 1, 1, 1, 1, 1}
91	91	{89, 2}	{11, 3}	57	514	01000000010_2	{2, 1, 1, 1, 1, 1, 1, 2, 1}	771	01100000011_2	{3, 1, 1, 1, 1, 1, 3}
92	92	{89, 3}	{11, 4}	58	516	01000000100_2	{2, 1, 1, 1, 1, 1, 2, 1, 1}	774	01100000110_2	{3, 1, 1, 1, 1, 3, 1}
94	94	{89, 5}	{11, 5}	59	520	01000001000_2	{2, 1, 1, 1, 1, 2, 1, 1, 1}	780	01100001100_2	{3, 1, 1, 1, 3, 1, 1}
96	96	{89, 5, 2}	{11, 5, 3}	60	522	01000001010_2	{2, 1, 1, 1, 1, 2, 2, 1}	783	01100001111_2	{3, 1, 1, 1, 5}
97	97	{89, 8}	{11, 6}	61	528	01000010000_2	{2, 1, 1, 1, 2, 1, 1, 1, 1}	792	01100011000_2	{3, 1, 1, 3, 1, 1, 1}
99	99	{89, 8, 2}	{11, 6, 3}	62	530	01000010010_2	{2, 1, 1, 1, 2, 1, 2, 1}	795	01100011011_2	{3, 1, 1, 3, 3}
100	100	{89, 8, 3}	{11, 6, 4}	63	532	01000010100_2	{2, 1, 1, 1, 2, 2, 1, 1}	798	01100011110_2	{3, 1, 1, 5, 1}

FIG. 2: Comprehensive list of even fibbinary related elements for $j = N \leq 100$ for even $n \leq 63$.



```
GoldenRatio
```

```
ϕ
```

```
N@ϕ
```

```
1.61803
```

```
τ = 1. / GoldenRatio
```

```
0.618034
```

```
{#, BaseForm[#, 2], BaseForm[grayCode@#, 2]} & /@ Range[0, 10]
```

```
 0     0_2     0_2
 1     1_2     1_2
 2    10_2    11_2
 3    11_2    10_2
 4   100_2   110_2
 5   101_2   111_2
 6   110_2   101_2
 7   111_2   100_2
 8  1000_2  1100_2
 9  1001_2  1101_2
10  1010_2  1111_2
```

```
AllCompositions@6
```

```
{{6}, {1, 5}, {5, 1}, {2, 4}, {4, 2}, {1, 1, 4}, {1, 4, 1}, {4, 1, 1}, {3, 3}, {1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}, {1, 1, 1, 3}, {1, 1, 3, 1},
 {1, 3, 1, 1}, {3, 1, 1, 1}, {2, 2, 2}, {1, 1, 2, 2}, {1, 2, 1, 2}, {1, 2, 2, 1}, {2, 1, 1, 2}, {2, 1, 2, 1}, {2, 2, 1, 1}, {1, 1, 1, 1, 2}, {1, 1, 1, 2, 1}, {1, 1, 2, 1, 1}, {1, 2, 1, 1, 1}, {2, 1, 1, 1, 1}, {1, 1, 1, 1, 1, 1}}
```

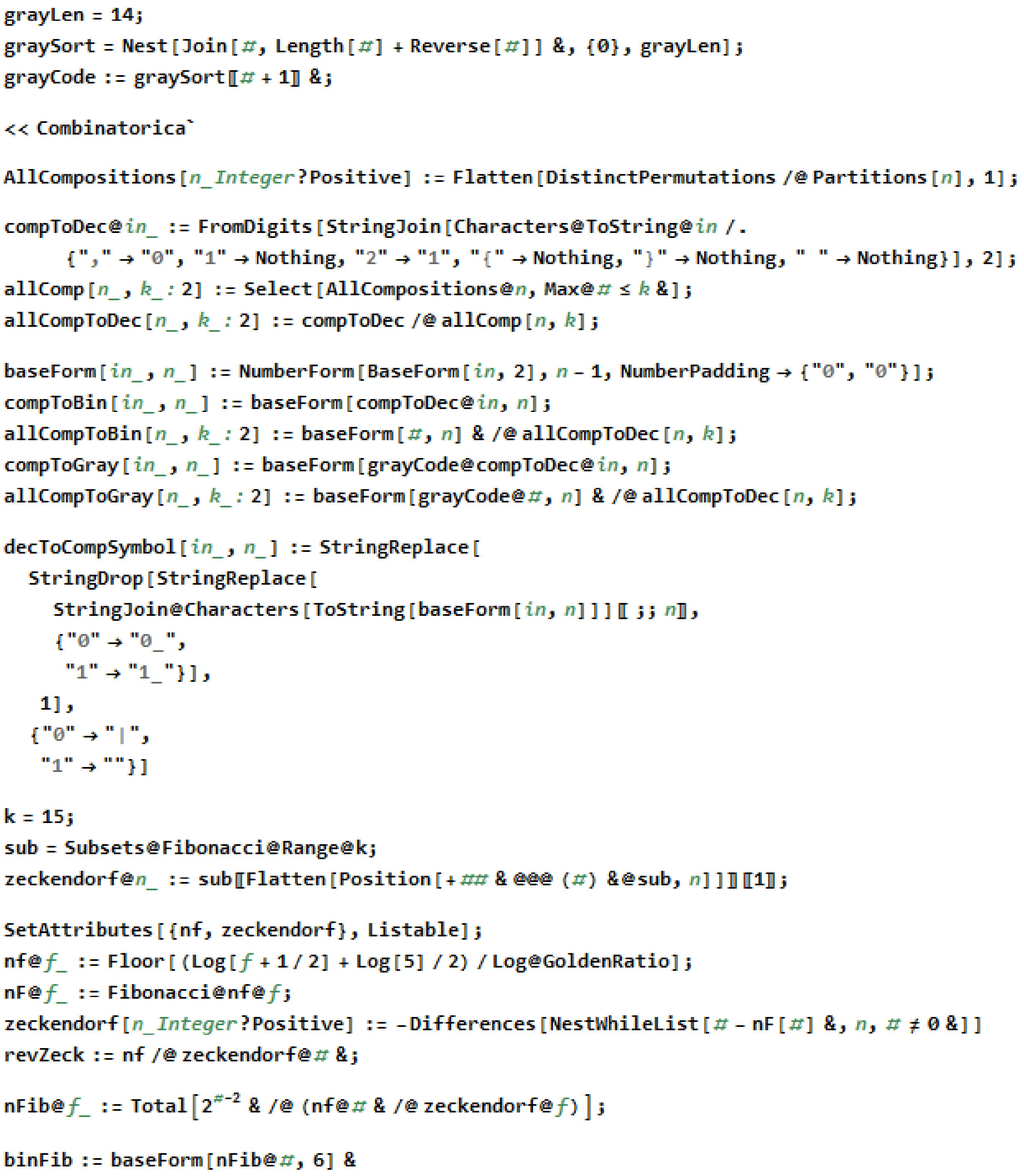
```
Row@{Column@{"Compositions of 6 {1,2}", "Decimal", "Binary", "Symbolic", "Gray Code: Decimal", "Gray Code: Binary", "Gray Code: Symbolic"},
  MatrixForm@{
    decToCompCount[#, 6] & /@ allCompToDec@6,
    allCompToDec@6,
    baseForm[#, 6] & /@ allCompToDec@6,
    decToCompSymbol[#, 6] & /@ allCompToDec@6,
    grayCode /@ allCompToDec@6,
    baseForm[#, 6] & /@ grayCode /@ allCompToDec@6,
    decToCompSymbol[#, 6] & /@ grayCode /@ allCompToDec@6}}
```

```
Compositions of 6 {1,2}  {2, 2, 2}  {1, 1, 2, 2}  {1, 2, 1, 2}  {1, 2, 2, 1}  {2, 1, 1, 2}  {2, 1, 2, 1}  {2, 2, 1, 1}  {1, 1, 1, 1, 2}  {1, 1, 1, 2, 1}  {1, 1, 2, 1, 1}  {1, 2, 1, 1, 1}  {2, 1, 1, 1, 1}  {1, 1, 1, 1, 1, 1}
Decimal                     21           5             9            10            17            18            20              1                2                4                8               16                0
Binary                   010101_2     000101_2      001001_2      001010_2      010001_2      010010_2      010100_2       000001_2         000010_2         000100_2         001000_2         010000_2         000000_2
Symbolic                 __|__|__     _|_|__|__     _|__|_|__     _|__|__|_     __|_|_|__     __|_|__|_     __|__|_|_      _|_|_|_|__       _|_|_|__|_       _|_|__|_|_       _|__|_|_|_       __|_|_|_|_       _|_|_|_|_|_
Gray Code: Decimal          31           7            13            15            25            27            30              1                3                6               12               24                0
Gray Code: Binary        011111_2     000111_2      001101_2      001111_2      011001_2      011011_2      011110_2       000001_2         000011_2         000110_2         001100_2         011000_2         000000_2
Gray Code: Symbolic      ______       _|_|____      _|___|__      _|______      ___|_|__      ___|___       _____|_        _|_|_|_|__       _|_|_|___        _|_|___|_        _|___|_|_        ___|_|_|_        _|_|_|_|_|_
```

```
{#, zeckendorf@#, nf@# & /@ zeckendorf@#} & /@ Range[89, 100]
```

```
 89     {89}      {11}
 90   {89, 1}   {11, 2}
 91   {89, 2}   {11, 3}
 92   {89, 3}   {11, 4}
 93 {89, 3, 1} {11, 4, 2}
 94   {89, 5}   {11, 5}
 95 {89, 5, 1} {11, 5, 2}
 96 {89, 5, 2} {11, 5, 3}
 97   {89, 8}   {11, 6}
 98 {89, 8, 1} {11, 6, 2}
 99 {89, 8, 2} {11, 6, 3}
100 {89, 8, 3} {11, 6, 4}
```

FIG. 3: *Mathematica*^{TM} example code output



```
grayLen = 14;
graySort = Nest[Join[#, Length[#] + Reverse[#]] &, {0}, grayLen];
grayCode := graySort⟦# + 1⟧ &;

<< Combinatorica`

AllCompositions[n_Integer?Positive] := Flatten[DistinctPermutations /@ Partitions[n], 1];

compToDec@in_ := FromDigits[StringJoin[Characters@ToString@in /.
      {"," → "0", "1" → Nothing, "2" → "1", "{" → Nothing, "}" → Nothing, " " → Nothing}], 2];
allComp[n_, k_: 2] := Select[AllCompositions@n, Max@# ≤ k &];
allCompToDec[n_, k_: 2] := compToDec /@ allComp[n, k];

baseForm[in_, n_] := NumberForm[BaseForm[in, 2], n - 1, NumberPadding → {"0", "0"}];
compToBin[in_, n_] := baseForm[compToDec@in, n];
allCompToBin[n_, k_: 2] := baseForm[#, n] & /@ allCompToDec[n, k];
compToGray[in_, n_] := baseForm[grayCode@compToDec@in, n];
allCompToGray[n_, k_: 2] := baseForm[grayCode@#, n] & /@ allCompToDec[n, k];

decToCompSymbol[in_, n_] := StringReplace[
  StringDrop[StringReplace[
    StringJoin@Characters[ToString[baseForm[in, n]]]⟦ ;; n⟧,
    {"0" → "0_",
     "1" → "1_"}],
   1],
  {"0" → "|",
   "1" → ""}]

k = 15;
sub = Subsets@Fibonacci@Range@k;
zeckendorf@n_ := sub⟦Flatten[Position[+## & @@@ (#) &@sub, n]]⟧⟦1⟧;

SetAttributes[{nf, zeckendorf}, Listable];
nf@f_ := Floor[(Log[f + 1 / 2] + Log[5] / 2) / Log@GoldenRatio];
nF@f_ := Fibonacci@nf@f;
zeckendorf[n_Integer?Positive] := -Differences[NestWhileList[# - nF[#] &, n, # ≠ 0 &]]
revZeck := nf /@ zeckendorf@# &;

nFib@f_ := Total[2^(#-2) & /@ (nf@# & /@ zeckendorf@f)];

binFib := baseForm[nFib@#, 6] &
```

FIG. 4: *Mathematica*^{TM} code snippets

[1] L. Lindroos, Electronic Theses and Dissertations.13: Integer Compositions, Gray Code, and the Fibonacci Sequence (2012).

[2] L. Lindroos, ArXiv e-prints math.CO/1812.02107 (2018), 1812.02107.

[3] J. G. Moxness, theoryofeverything.org blog post: Integer Compositions, Gray Code, and the Fibonacci Sequence (2018).