Proceedings of The National Conferences
On Undergraduate Research (NCUR) 2017
University of Memphis, TN
Memphis, Tennessee
April 6-8, 2017

Explorations of Finger Games

Daniel J. Tjie
Mathematics
Ithaca College
Ithaca, New York

Faculty Advisor: Dr. John Rosenthal

Abstract

Finger Games, a topic in number theory, study certain 1-1 operations on sequences of 0's and 1's of length $2f$ - viewing the left and right half as fingers of adjacent hands interpreted using what is called Gray Code. In particular, Finger Games involve the operation of counting in Gray Code by an even number C, alternating hands. As the operations are 1-1, they divide the sequences into orbits. This paper focuses on the even parts of orbits for C = 6, 10, 14, and 18.

Keywords: Number Theory; Gray Code

1. Introduction

1.1 Gray Code with Odd and Even Moves

Finger Games uses a notation system called **Gray Code**[1] (different than binary) for counting using 0 and 1 bits. Every nonnegative integer can be represented by a unique sequence of 0's and 1's. Adding 1 to an even number changes the bottom bit, called an **odd move**, as this results in an odd number. Adding 1 to an odd number changes the bit above the lowest 1-bit, called an **even move**, as this results in an even number. Both moves can be applied to all nonnegative numbers with the exceptional case $\boldsymbol{E}(0)= 0$. $\mathbf{E}\circ\mathbf{E}$ and $\mathbf{O}\circ\mathbf{O}$ are both the identity function, $\boldsymbol{I}$.

1.2 Useful Information

1. 2^k is the smallest $k + 1$ bit number with the form 11 followed by $k - 1$ 0's
2. $2^k - 1$ is the largest k bit with the form 1 followed by $k - 1$ 0's
3. An even number has an even number of 1's and an odd number has an odd number of 1's

2. Finger Games

With Finger Games, there are two sequences of Gray Code of the same length, which can be divided into a **right hand** and **left hand** facing each other together forming a **configuration**. The bottom bits of each hand are the pinkies and the top bits of each hand are the adjacent thumbs. The individual places, representing a bit, are called **fingers**. A **down finger** represents a 0 and an **up finger** represents a 1.

For example, a 7 read backwards (001 with the rightmost 1 being the top bit) on the left and a 4 (110 with the leftmost 1 being the top bit) on the right give the combined configuration 001110.

2.1 Odd and Even Moves, Left and Right Hands

Even and odd moves can be applied to configurations on the combined hand.

$\boldsymbol{O_R}$- odd move on the right, which changes the rightmost bit of the combined sequence.

$\boldsymbol{O_L}$- odd move on the left, which changes the leftmost bit of the combined sequence.

$\boldsymbol{E_R}$- even move on the right, which changes the bit to the left of the rightmost 1.

$\boldsymbol{E_L}$- even move on the left, which changes the bit to the right of the leftmost 1.

Just as there is an exception for $\boldsymbol{E}$, there are several exceptions for $\boldsymbol{E_R}$ and $\boldsymbol{E_L}$.

$\boldsymbol{E_R}$ does nothing to (0 … 0) and (1 … 0) and $\boldsymbol{E_L}$ does nothing to (0 … 0) and (0 … 1).

These moves $\boldsymbol{O_R}$, $\boldsymbol{O_L}$, $\boldsymbol{E_R}$, and $\boldsymbol{E_L}$ are 1-1 and onto (as they are self-inverses).

Table 1: Gray Codes for Small Integers.

Integer	**Gray Code**	**Notes**
0	0	
1	1	The bottom bit was changed from 0 to 1.
2	11	The bit to the left of the bottom 1 was changed from 0 to 1. 2^k where $k = 1$ is the smallest $k + 1$ bit number.
3	10	The bottom bit was changed from 0 to 1.
4	110	The bit to the left of the bottom 1 was changed from 0 to 1. 2^k where $k = 2$ is the smallest $k + 1$ bit number.
5	111	The bottom bit was changed from 0 to 1.
6	101	The bit to the left of the bottom 1 was changed from 0 to 1.
7	100	The bottom bit was changed from 0 to 1. $2^k – 1$ where $k = 3$ is the largest k bit number.
8	1100	The bit to the left of the bottom 1 was changed from 0 to 1. 2^k where $k = 3$ is the smallest $k + 1$ bit number.

2.2 Counting by C

Let $\boldsymbol{f}$ denote the number of fingers in each hand. Let $\boldsymbol{C}$, an even number (by which one counts) be a lot less than 2^f, usually $< 2^{f/2}$. Counting by C consists of repeatedly applying C moves (alternating between odd & even moves) on one hand and then on the other hand repeatedly alternating between the two hands.

2.3 Positions & Orbits

As the Counting by C function of each hand alternates between O_R and E_R (or O_L and E_L), the counting by C function is also 1-1 and onto. As there are finitely (2^{2f}) many **configurations**, each position will eventually return to itself, thus creating **orbits**.

A position is called an **even position (**or **odd position)** if the fingers combined between the two hands represent an even number (or odd number, respectively). So a position is even when either both hands are even or both hands are odd. An orbit is called an **even orbit** if all its positions are even; an **odd orbit** if all its positions are odd; and is called a **mixed orbit** if it has even and odd positions.

2.4 Related Work

Past students worked with Professor Rosenthal on Finger Games. Brittany Rose[2] in 2012 computed the even parts of orbits for C = 4 and 8 and had most of the ideas for 2^n under the guidance of Professor Rosenthal. This paper discusses C = 6, 10, and 14 in full and partially discusses C = 18. The results for each values of C are discovered by a long sequence of results that build on previous results – both of previous students and mine.

2.5 Useful Lemmas

Let $\boldsymbol{d = 2^f \textbf{ mod } C}$ and $\boldsymbol{u = C - d}$. This notation is utilized at the beginning of section 3.

2.5.1 lemma 1 (mod 6)

For $f \geq 1$:
If f = even, then 2^f = 4 mod 6. So d = 4; u = 2
If f = odd, then 2^f = 2 mod 6. So d = 2; u = 4

2.5.2 lemma 2 (mod 10)

For $f \geq 1$:
If f = 0 mod 4, then 2^f = 6 mod 10. So d = 6; u = 4
If f = 1 mod 4, then 2^f = 2 mod 10. So d = 2; u = 8
If f = 2 mod 4, then 2^f = 4 mod 10. So d = 4; u = 6
If f = 3 mod 4, then 2^f = 8 mod 10. So d = 8; u = 2

2.5.3 lemma 3 (mod 14)

For $f \geq 1$:
If f = 0 mod 3, then 2^f = 8 mod 14. So d = 8; u = 6
If f = 1 mod 3, then 2^f = 2 mod 14. So d = 2; u = 12
If f = 2 mod 3, then 2^f = 4 mod 14. So d = 4; u = 10

2.5.4 lemma 4 (mod 18)

For $f \geq 1$:
If f = 0 mod 6, then 2^f = 64 mod 18 = 10 mod 18. So d = 10; u = 8
If f = 1 mod 6, then 2^f = 2 mod 18. So d = 2; u = 16
If f = 2 mod 6, then 2^f = 4 mod 18. So d = 4; u = 14
If f = 3 mod 6, then 2^f = 8 mod 18. So d = 8; u = 10
If f = 4 mod 6, then 2^f = 16 mod 18. So d = 16; u = 2
If f = 5 mod 6, then 2^f = 32 mod 18 = 14 mod 18. So d = 14; u = 4

We denote the number of cases in the lemmas as $L(C)$.

2.6 Both Hands Even

If both hands are even, then counting by C:

E1) On the right increases the even value n on the right to $n + C$ unless an "overflow" occurs, that is, $n + C \geq 2^f$. This is called a **top interaction of the right on the left** (changing the top bit on the left, that is, applying an odd move on the left read backwards).

E2) On the left increases the even value n on the right to $n + C$ unless an "overflow" occurs, that is, $n + C \geq 2^f$. This is called **a top interaction of the left on the right** (changing the top bit on the right).

When both hands are even, a top interaction changes both hands to odd.

2.7 Both Hands Odd

If both hands are odd, then counting by C:

O1) On the right decreases the odd value n on the right to $n - C$ by C unless an "underflow" occurs, that is, $n < C$. This is a bottom interaction from the right on the left (changing the bit below the top most 1-bit on the left, that is, applying an even move to the left read backwards).

O2) On the left decreases the odd value n on the right to $n - C$ by C unless, unless an "underflow" occurs, that is, $n < C$. This is a bottom interaction from the left on the right (changing the bit below the top most 1-bit on the right).

When both hands are odd, a bottom interaction changes both hands to even.

2.8 Multistep

If both hands are even, continue to add C alternately to each hand (**upsweep)** until one hand reaches $2^f - 1$, then changing the top bit on the other hand (**top interaction)**. Then when both hands are odd, continue to subtract by C alternately on each hand (**downsweep)** until one hand 0, then changing the bit above the lowest 1-bit on the other hand (**bottom interaction)**. An upsweep, top interaction, downsweep, and bottom interaction is called a **multistep**. Theorems 1A and 1B describe these more fully.

The underlined entry indicates the side that counts first.

2.9 Mixed Orbits

The mixed orbits of $(e, \underline{0})$ for $0 \leq e < C$ where e is even. This is mixed because the orbit contains odd and even positions. The previous odd position is $(\underline{C - e - 1}, 0)$.

2.9.1 even orbits

There are two types of Even Orbits: Short Even Orbits and Long Even Orbits to be described later.

2.9.2 rough statement of theorem 1a

If $d \leq a$, then after a multistep $(a, \underline{b})$ is sent to $(a - d, \underline{L_2(b - d)})$ where L_2 is counting by 2 from the top, i.e. changing the top bit and then changing the bit below the topmost 1.

2.9.3 rough statement of theorem 1b

If $0 \leq a < d$, then after a multistep $(a, \underline{b})$ is sent to $(a + u, \underline{L_2(b + u)})$ where L_2 is as above.
These theorems have exceptions.

C: In case a) when $d \leq a < C$ and $b < d$ since then $b - d < 0$.

D: In case b) when $0 \leq a < d$ and $b + u \geq 2^f$.

E: In case b) when $0 \leq a < d$ and $b + u = 2^f - 2$, so $L_2(b + u) = 2^f$.

2.9.4 correct statement of theorem 1[3]

A: If a and b are even, $d \le a < C$, and $d \le b < 2^f$ then, after a multistep (a, $\underline{b}$) is sent to ($a - d$, $\underline{L_2(b - d)}$) where L_2 is as above.

B: If a and b are even, $0 \le a < d$, and $0 \le b < 2^f - u$ then, after a multistep (a, $\underline{b}$) is sent to ($a + u$, $\underline{L_2(b + u)}$) where L_2 is as above.

C: If $0 \le a \le d$, and $b < d$, then after a multistep (a, $\underline{b}$) → ($\underline{L_2(a - d)}$, $b + u$). So in exception C, a lead change occurs, that is, the leading hand of the start of successive multistep changes.

Lead changes or the end of an even part also occur in Exceptional Cases D and E.

3. Drum Roll, Tops, & Bottoms

Let b = # of bits of C.

Theorem 1A and 1B imply the left hand repeatedly goes thru a pattern of ups (counting up by u) and downs (counting down by d) starting and ending at 0, which is called a **drum roll**. The right hand can be broken into a **bottom** consisting of b bits and a **top** consisting of the remaining $g = f - b$ bits. The bottom of the right imitates the drum roll, that is it follows the pattern of the drum roll of the left hand (that is, counting up by u or down by d). More precisely, if the bottom is even, add u or subtract d and if the bottom is odd, subtract u or add d. Each stage on the right also counts by 2 on the top **read backwards** at each stage; that is to say when the top is even, add 2, and when the top is odd, subtract 2. All actions involving the top involve it being read backwards.

Table 2: Examples of Drum Rolls

Case	**Drum Roll**
k = odd for $C = 6$ ($u = 2$; $d = 4$)	0 → 2 → 4 → 0
k = 2 mod 4 for $C = 10$ ($u = 6$; $d = 4$)	0 → 6 → 2 → 8 → 4 → 0
k = 2 mod 3 for $C = 14$ ($u = 10$; $d = 4$)	0 → 10 →6→ 2 → 12 → 8 → 4 → 0
k = 2 mod 6 for $C = 18$ ($u = 14$; $d = 4$)	0 → 14 → 10 → 6 → 2 → 16 → 12 → 8 → 4 → 0

As noted above, the bottom imitates the drum rolls. For example, for $C = 14$ with $f = 2$ mod 3, if the bottom starts at 5 then in a drum roll, the bottom goes:

5 → 4 → 0 → 3 → 6 → 2 → 1 → 5. (As with bottom underflows as described below at the transitions from odd to even or even to odd).

3.1 Top and Bottom Overflows and Underflows

3.1.1 top overflow

A Top Overflow occurs after the even top (of the hand) counts up by 2 reaching $2^g - 2$, adding 1 gives $2^g - 1$, then changes the top-most bit of the bottom.

3.1.2 top underflow

A Top Underflow occurs after the odd top (of the hand) counts down by 2 reaching 1, subtracting 1 gives 0, then changes the bit below the top-most 1-bit of the bottom.

3.1.3 bottom overflow

A Bottom Overflow occurs after bottom reaches $2^b - 2$, adding 1 gives $2^b - 1$, then changes the top-most bit of the top. After the bottom underflow the counting on the bottom by *u* or *d* is completed, and then as usual one counts by 2 on the top.

3.1.4 bottom underflow

A Bottom Underflow occurs after the bottom reaches 1, subtracting 1 reaches 0, subtracting 1 then changes the bit below the top-most bit on the top read backwards. After the bottom underflow the counting on the bottom by *u* or *d* is completed, and then as usual one counts by 2 on the top.

3.2 Top & Bottom Overflow & Underflow Analysis

The most common overflows and underflows are by the bottom; the bottom underflow needs the most analysis.

Bottoms overflows and underflows are most common because in a multistep the bottom mirrors the ups and downs of the drum roll and can change by more than 2 while the top only counts by 2. In addition, the top is typically longer than the bottom so it is less likely for the top to reach an overflow or underflow.

In a drum roll, often bottom overflows occur in pairs, which "cancel" (and so can be ignored) as they change the same bit on the top. Exceptions occur if a top overflow or top underflow also occurs within a drum roll. Bottom underflows also often cancel but only when the position of the top-most 1-bit on the top hasn't changed. When the bottom overflows or underflows cancel within a drum roll, the top just changes by C (add C to the top if the top is even and subtract C from the top if the top is odd). With bottom overflows, a separate analysis is needed when a top overflow or underflow also occurs within a drum roll.

4. String Pictures

A String Picture is a visual representation of where a current position will go after many drum rolls when the bottom is an odd number $< C$ (when the other hand is 0) and so has bottom underflows. The highest level uses all bits for the top. Each subsequent level needs one fewer bit than the level above and levels repeat every $L(C)$ rows. Transitions from one level to another occur when the number of bits needed to write the top increases or decreases by one. By the number of bits to write the top we mean the bit number of the top-most 1 of the top. Henceforth we call this the length of the top. In a drum roll, bottom underflows cancel if and only if the number of bits to write the top doesn't change. The arrows show the transitions up or down from any level. This is because levels repeat every $L(C)$. For example, with $C = 14$, four levels are enough to show the repetitions although for other reasons, five levels may be preferable.

Looking back at the lemma section, for:

- Counting by 6, there are 2 + 1 = 3 levels
- Counting by 10, there are 4 + 1 = 5 levels
- Counting by 14, there are 3 + 1 = 4 levels (but prefer 5 levels)
- Counting by 18, there are 6 + 1 = 7 levels
    - We will see that with Counting by 18 we actually need $(6 \cdot 2) + 1 = 13$ levels

We use two types of string pictured called **Rising String Pictures** for bottoms an odd number with $0 < n < \frac{C}{2}$ and **Falling String Pictures** for bottoms and odd number with $\frac{C}{2} < n < C$. In both pictures the top-most level is $g = f - b$. Each level shows all tops of a particular length (for example $2^{1 \bmod 3}$). Each level has two rows, the upper row (for example in Figure 1 below $2^{1 \bmod 3} - 1$) showing the odd number where arrows enter the level (for example $2^{0 \bmod 3} + 1$) and the lower row showing where arrows leave the level by changing the length of the top.

The transition from the upper row to the lower row is by repeated subtractions of *C*. The values for the lower row are found from the upper row using the appropriate case of the lemma for that value of *C*. The only difference between Rising and Falling String Pictures is how different levels are placed relative to each other.

In Falling String Pictures, the levels are lined up so that the entry in the top level is *C* less than the entry in the bottom of one level above. In Rising String Pictures, the levels are lined up so that the entry in the nonpower of 2 part of the entry in the top row of a level is *C* less than the nonpower of 2 part of the entry in the bottom of one level below. For example, in Figure 1, $2^{1 \bmod 3} - 1$ is placed above $2^{2 \bmod 3} + 13$ (the -1 is 14 below + 13 as $C = 14$). These layouts were chosen to make paths that don't reverse levels look straight, and to make paths swerve (change columns) when the levels go up or down in different stages of the path.

5. Discussion of Finger Games Results

Lots of results of Fingers Games can be derived from String Pictures such as the one illustrated below in Figure 1.

5.1 String Pictures Follow Path

As shown above (and with any string picture), there are "straight" paths up, "straight" paths down, swerve paths, and cycles. To reiterate, "straight" paths move in one direction always increase or decrease in length. Swerve paths will increase and decrease in length (multiple times). Sometimes in a drum roll, the length changes by more than 1-bit, which requires more analysis as explained below. Cycles give rise to even orbits, which we call short even orbits, since positions return to themselves. This is shown on the string pictures by closed "polygons." The length will increase and decrease by only a few. Cycles usually come in pairs because of a "duality" between a bottom of *Ó* and a bottom of *C* - *Ó* where *Ó* is odd < *C*.

Since levels of a string picture repeat, in the repetitions of levels the strands have the same behavior. Let's take a look at $C = 14$ for $f = 2 \bmod 3$ with the bottom being 5 and specifically Cycle 5A with the lines called **strands**. Here are two observations. First all of the strands in 5A fit into the entire picture and show where each path goes. Second, since cycles repeat, selecting a starting position will eventually return to that position.

We write (*a*, [*k*,*m*]) to mean that *a* is the left hand, *k* is the top and *n* is the bottom of the right. Take a look at the topmost left cell that prints $2^{0 \bmod 3} + 7$. After one full drum roll, we end up at (0, [6, 10]), which we denote as [6, 10] for short.

f = 2 mod 3 2^f = 4 or 18 mod 14 d = 4 u = 10

Drum roll: 0 → 10 →6 →2 → 12 →8 →4 →0
Bottom roll: 5 →4→0→ 3 →6→2→ 1 → 5
Levels up with 1, 3, 7, and 9; Levels down with 5, 11, and 13

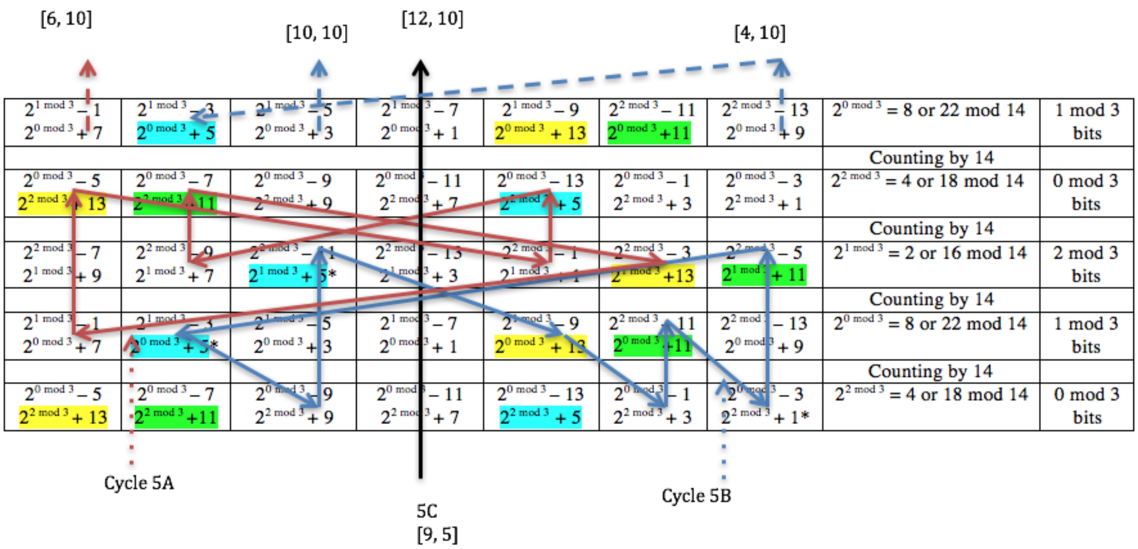

Figure 1: Bottom of the Right 5

We use 5 levels rather than 4 levels to make the cycles easier to see.

When a position with small length is going to change length by more than 1-bit, a drum roll needs to be done to verify positions do indeed follow the path. At low levels, the entries may not have the correct length. These correspond to a drum having more than one length change. For example, $2^3 + 9 = 17$ has length 5, not 4. After studying all situations where this happens with $C = 6$, $C = 10$, $C = 14$, and $C = 18$, we found whenever there is a "culprit," that drum roll will skip all culprit positions and bring one to the next correct position on the strand.

5.2 Upper-Level of Strings

The connection you want to make is that three levels below the topmost left cell is also $2^{0 \bmod 3} + 7$. Suppose that is our initial position. Then as we follow Cycle 5A, we will end up at $2^{0 \bmod 3} + 7$. So, if we start at [6, 10] above, we will eventually return to [6, 10]. The question is how to we get back without the upper levels? Further analysis examines what happens above the top of the string picture. Not only does such analysis show that Cycle 5A above this top, but it also shows how strings for odd bottom value often transition to strings for perhaps a different value of the bottom.

5.3 Short Even Orbits & Long Even Orbits

With **Short Even Orbits**, lengths of the top do not change by much throughout the orbit, as in, the number of bits needed to write a certain number only shortened or increased by a few bits (in the cycles in the String Pictures).

Many strands occur in the mixed orbits. But strands not occurring in mixed orbits must be part of even orbits that do up or down through virtually all levels. Hence, such even orbits are called **Long Even Orbits.**

6. Broad Overview of Results

Table 3: Summary of Orbits

	C = 6	**C = 10**	**C = 14**	**C = 18**
Short Even Orbits	Two short even orbits arising from cycles with bottoms of 1 and 3 for f = 0 and 1 mod 2	No cycles and so no short even orbits	Four short even orbits arising from cycles with bottoms of 5 and 7	18 short even orbits for f = 2 mod 6 24 short even orbits for f = 1 and 3 mod 6 28 short even orbits for f = 4 mod 6 32 short even orbits for f = 5 mod 6 26 short even orbits for f = 0 mod 6
Long Even Orbits	No long even orbits	Two long even orbits only for f = 2 mod 4	Two long even orbits only for f = 1 mod 4	No long even orbits for f = 2 mod 6
Mixed Orbits	Three mixed orbits for f = 0 and 1 mod 2	Five mixed orbits for f = 0, 1, 2, & 3 mod 4	Seven mixed orbits for f = 0, 1, & 2 mod 3	15 mixed orbits for f = 2 mod 6

It may appear that the number of mixed orbits are half of what the C value is but this is not true for C = 18 as there are 15 mixed orbits (not 9 mixed orbits). C = 18 has more mixed orbits than expected because in string pictures several strands will pass through 12-levels before returning to the same column. This implies there are actually two paths – one for f = 2 mod 12 and one for f = 8 mod 12 (each of which has 9 mixed orbits).

7. Future Work

My research includes finding all even parts of orbits for a selected f and C. At the moment, only C = 18 is completed for f = 2 mod 6. For f = 0, 1, 3, 4, and 5 mod 6. I have so far only obtained the string pictures and resulting short even orbits. I have not yet studied even parts of mixed orbits and long even orbits (if any). Finding mixed orbits is my next goal.

For future students who work on this research project, other possible ideas to investigate and look into are: finding odd parts; predicting when lead changes will occur before encountering one; creating a computer program that quickly calculates the orbits; and seeing if the even parts of orbits for 2C can be found from those for C.

1. Doran, Robert W. "The Gray Code." *J. UCS* 13.11 (2007): 1573-1597
2. Rose, Brittany, *Gray Code and Finger Games: Counting by 4 and 8*, 2011 (Unpublished)
3. Rosenthal, John, *Results of Multisteps*, 2014 (Unpublished)

Acknowledgments: I am grateful to have worked under the guidance of Dr. John Rosenthal, Department of Mathematics, at Ithaca College in Ithaca, NY. As my faculty mentor on this project, he patiently reviewed my manuscript countless times and provided valuable feedback, direction, and suggestions for this journal publication. He also guided me through my presentations at the Whalen Academic Symposium in April 2016 & 2017 at Ithaca College, MathFest in July 2016 in Columbus, OH, the National Conference of Undergraduate Research (NCUR) in April 2017 at the University of Memphis in Memphis, TN, and my Mathematics Honors Presentation in April 2017 at Ithaca College. I would also like to thank the entire Department of Mathematics at Ithaca College for providing me with the knowledge, training, and support. Lastly, I would like to recognize the Summer Scholars Program, sponsored by the School of Humanities and Sciences, at Ithaca College for funding my research during the summer of 2016.