Math 319NT Fall 2024 **Name: Aby Soumare** 0cm]
Lab 1: Differences and Sums of Squares + Intro to Magma
Collaborators:

Introduction

In this lab, we were tasked with two problems to play with:

Question 0.1 *Given an integer n, find all ways to express n as a difference of two squares.*

Question 0.2 *Given an integer n, find all ways to express n as a sum of two squares.*

To answer these problems, we used the language Magma to help us visualize our work. Along with that, we used a .txt file, a .tex template, and a worksheet given to us by Professor Tori Day to help us through the process.

Analysis

Exercise 0.1 *Was a list of four quick problems Professor Tori Day tasked us with solving:*

- *Compute the floor of 7*
- *Compute the floor of 7.5*
- *Check to see if 1837 is a square*
- *Check to see if 5184 is a square*

The answers to these problems are:

- Floor(7) = 7
- Floor(7.5) = 7
- IsSquare(1837) - NO
- IsSquare(5184) - NO

Exercise 0.2 *Was yet another list of four problems tasked by Professor Tori Day:*

- *An expression for 17 as the difference of two squares*
- *An expression of 1837 as the difference of two squares*
- *An expression of 6 as a difference of two squares*
- *An expression of your favorite integer (8) as the difference of two squares*

The answers to these problems are:

- DiffSquares(17) $= 9^2 - 8^2$
- DiffSquares(1837) $= 919^2 - 918^2$
- DiffSquares(6) $=$ NONE
- DiffSquares(8) $= 3^2 - 1^2$

Exercise 0.3 *We were tasked to use the count function to find the "number of ways to write your favorite positive number as a difference of two squares":*

```
DiffSquares(17);
```

And this ended up being simply 1.

Exercise 0.4 *We were tasked to use DiffSquares() to finish writing the code for the template Professor Day gave us. This is the completed code:*

Code 0.3 *Completed for loop*

```
for n in [1..100] do
    print n, DiffSquares(n);
end for;
```

With this code, we were able to discover that 96 has the most expressions as a difference of 2 squares.

Exercise 0.5 *Now we are tasked with having to explain what each line of code is doing in this for loop:*

Code 0.4 *Commented for loop*

```
for n in [1..100] do %we are taking all numbers 1 through 100 and doing something
    print n, DiffSquares(n); %we are first printing the number
    %we are modifying, and then printing their DiffSquares if it exists
end for; %ending the loop
```

Exercise 0.6 *Now we are tasked with having to explain what each line of code is doing in the function SumSquares() given to us by Professor Day:*

Code 0.5 *Commented SumSquares() function:*

```
function SumSquares(n) %start function
    S:={}; %creating a list
    for x in [0..Floor(SquareRoot(n/2))] do %taking all n
    %on the line sqrt(n/2) rounded down
        t,y:=IsSquare(n-x^2); %checking to see if the
        %number we are observing is square
        if t then %if it is true
            S:=S join {[x,y]}; %add to the list and save the x,y
            end if;
        end for;
    return S; %return the list
end function;
```

Exercise 0.7 *Now we are to use the SumSquares() function to find the following:*

- An expression for 20 as a sum of 2 squares: [2,4]
- An expression of 1837 as a sum of 2 squares: []
- An expression of 6 as a sum of two squares: []
- An expression of 8 as a sum of two squares: [2,2]

Exercise 0.8 *Now we are to use SumSquares() to determine which integer between 1 and 100 has the most sum squares:*

- 25: 2
- 50: 2
- 65: 2
- 85: 2
- 100: 2

Exercise 0.9 *Finally, we were to find which prime numbers between 1 and 100 can be expressed as a sum of squares:*

- 2
- 5
- 13
- 17
- 29
- 37
- 41
- 53
- 61
- 73
- 89
- 97

Conjecture 0.6 *Note that there are 12 items in this list of numbers. Also note that there are 25 prime numbers between 1 and 100. As one may observe, if we were to add up how many numbers prime numbers and Floor this number, we are going to get the number of prime numbers that can be expressed as a sum of squares. This also applies to the primes greater than 100 and less than 200.*

Future Directions

There are many unanswered questions that have come up in this lab, and here are a couple of the questions that we noted down:

- How would one predict which prime numbers can be expressed by a sum of two squares?
- Is there a way to create a for loop that sums squares instead of creating a function?
- Is there a way to create a function that prints the number(s) with the most expressions of sums of squares without having to manually find the answer?

Appendix

The following page is all the code used for this lab:

Code 0.7 *Difference of Squares Function Given by Prof Day:*

```
function DiffSquares(n)
    S := {};
    for x in [0..Floor(n+1/2)] do
        t,y := IsSquare(x^2-n);
        if t then
            S := S join {[x,y]};
        end if;
    end for;
return S;
end function;
```

Code 0.8 *Summation of Squares Function Given by Prof Day:*

```
function SumSquares(n)
    S:={};
    for x in [0..Floor(SquareRoot(n/2))] do
        t,y:=IsSquare(n-x^2);
        if t then
            S:=S join {[x,y]};
            end if;
        end for;
    return S;
end function;
```

Code 0.9 *Completed For Loop, Template Given by Prof Day:*

```
    for n in [1..100] do
        print n, DiffSquares(n);
        end for;
```

Code 0.10 *For Loop for Ex. 0.8*

```
    for n in [1..100] do
        print n, #SumSquares(n);
        end for;
```

Code 0.11 *For Loop for Ex. 0.9*

```
    for n in PrimesUpTo(100) do
        print n, #SumSquares(n);
        end for;
```