

Coding Hail Stones

Student Activity

7 8 9 10 11 12



Note: This activity is also available in a non-coding version, where students use existing programs to do a similar exploration

Introduction

The Hail Stone sequence is produced by a relatively simple formula. What makes this sequence important and popular is Collatz's conjecture: "Start with any number and the sequence will eventually produce a one (1)." This statement by Collatz is referred to as a conjecture because it has never been proven. Trying lots of numbers doesn't prove the statement, it simply shows that all the numbers tested so far end up with a one. Finding a sequence that seems to go on for ever doesn't prove Collatz wrong, unless it has produced a loop. Really long sequences are exactly that, really long. His conjecture could be proved using a mixture of number theory and algebra, but despite numerous efforts, no one has succeeded in such a proof. So what are we looking for in this investigation? Patterns, generating ideas based on observable patterns. A powerful problem solving technique is to throw some numbers into the mix and pay careful attention to what comes out.



So why is the conjecture referred to as "The Hailstone Problem"? Hail stones are formed when high level winds carry water droplets upwards where the air temperature is cold enough to freeze water. The water droplets become frozen, as they continue to go up and down through these high altitudes they get bigger and heavier, and eventually fall to the ground. The numbers in the sequence go up and down, and according to Collatz, will eventually come tumbling down to one!

The Rule

Start with any number:

- If it is odd, multiply it by three then add one. Odd ... $\times 3 + 1$
- If it is even, divide by two. Even ... $\div 2$

Repeat this odd/even process until you reach 'one'.

Example:

Starting Number: 11
Since 11 is odd: $11 \times 3 + 1 = 34$
Since 34 is even: $34 \div 2 = 17$
Since 17 is odd: $17 \times 3 + 1 = 52$
Since 52 is even: $52 \div 2 = 26$

Question: 1.

Write down the remaining numbers in the sequence until one (1) is reached.

11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

Question: 2.

Write down the sequence for the set of numbers starting with 3.

3, 10, 5, 16, 8, 4, 2, 1

Question: 3.

Without making any more calculations, write down the sequence for the set of numbers starting with 5. Explain how you were able to write this sequence down without any further calculations.

5, 16, 8, 4, 2, 1 - The sequence is a part (subset) of the sequence produced for 3.

It is important for students to realise that the rule produces a definitive sequence for every number.

Question: 4.

Write down the sequence for the numbers listed in the table below.

(Some of the sequences have already been completed for you.)

The numbers below have been selected to reinforce the notion that it is possible to identify specific sequences of numbers, particularly the trailing part of the sequence. 10, 5, ... 2, 1. Do all sequences finish this way? Write a sequence that doesn't finish this way, refer to the inquiry questions.

Start	Sequence	Number of Terms
6	3, 10, 5, 16, 8, 4, 2, 1	9
48	24, 12, 6, 3, 10, 5, 16, 8, 4, 2, 1	12
13	40, 20, 10, 5, 16, 8, 4, 2, 1	10
7	22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1	17
9	28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1	20
29	88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1	19
53	53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1	12

Question: 5.

Some sequences are surprisingly short; others can be extraordinarily long, particularly considering their starting value. The sequence for 53 contains just 12 terms, the sequence for 54 contains 113. How many terms would be in the sequence for the starting value of 864?

864, 432, 216, 108, 54 ... (111 more terms). Total terms: 117. *This question reinforces the notion that you do not have to reproduce the entire sequence once it arrives at a value that has previously been included in a sequence.*

Automating the Process

Testing numbers can be a very time consuming process. In this investigation we are interested in exploring patterns that may help explain why the sequence seems to always end at one. Computers are very quick at making programmed decisions and performing calculations. In this section you will write a computer (calculator) program to perform the calculations very rapidly. A few commands must be learnt before writing the program. The following questions will help understand these commands.

Question: 6.

Record the result for each of the following calculations. The MOD command can be retrieved from the **Number** menu in the calculator application, followed by **Number Tools** then **Mod**; alternatively the word Mod can be typed directly using the keypad. Once the command has been used it can be copied (Ctrl + C) and pasted (Ctrl + V) ... and re-pasted (Ctrl + V) for each subsequently use.

- a) $\text{Mod}(25,2) = 1$ b) $\text{Mod}(13,2) = 1$ c) $\text{Mod}(12, 2) = 0$
 d) $\text{Mod}(40,2) = 0$ e) $\text{Mod}(9,2) = 1$ f) $\text{Mod}(100, 2) = 0$
 g) $\text{Mod}(1,2) = 1$ h) $\text{Mod}(1008,2) = 0$ i) $\text{Mod}(147, 2) = 1$

Question: 7.

Explain what $\text{Mod}(n,2)$ does in terms of the number n .

Finds the “remainder” of n when divided by 2, so the answer is 1 for odd numbers and 0 for even numbers.



If you are completely new to programming, try the introductory programming lessons on the Texas Instruments website: 10 minutes of coding.

<https://education.ti.com/en-GB/aus-nz/product-resources/ti-codes>

Programming

The entire program contains less than 10 lines of code. The real decision making process occurs with the testing statement “IF ... THEN ... ELSE”.

Request n	To test a single number (n) the user will be prompted to input a number. The ‘Request’ command seeks an input.
WHILE n > 1	This line assumes the sequence will eventually end up at one, so the instructions are repeated while n is bigger than 1.
IF n is odd THEN	Decision time, need to automatically determine if a number is odd. To achieve this: $\text{Mod}(n,2)$ will be used.
$n := 3 \times n + 1$	This is not an algebraic statement. If n is odd, value of n changes according to the rule. “ := ” is read as ‘assigned equal to’.
Else	If the number is not odd... it must be even.
$n := n \div 2$	This is not an algebraic statement. The value of n changes according to the rule for even numbers.
Endif	Programming command representing the end of the decision making process.
Disp n	For the benefit of the user, the new value of n is displayed so that it is possible to see the entire sequence.
EndWhile	This is the end of the loop that performs the calculations, and will also be the end of the program once n reaches 1.

On the calculator application, use the **[Menu]** key then select **Functions & Programs**, followed by **Program Editor** then **New**.

Type the program name: Hail

Press **[Enter]** then arrow down to the first space in the program.

The first step in the program is to request an input value. From the **[Menu]** select **I/O** (Input/Output) followed by **Request**. The request is best supported with a word prompt.

Type: "Number" , n

The quotation marks can be accessed by selecting the **[?!]** key. The variable n is short for number and will hold the initial number, but will change as soon as the program executes. The second step in the program is to start a loop which will continue while the value of n is greater than 1. From the **[Menu]** select **Control** followed by **While ...EndWhile**.

Type: $n > 1$

The ' $>$ ' sign is located 'under' the [=] key and can be accessed by pressing **Ctrl + =**

At this point the decision algorithm is involved. The command $\text{Mod}(n,2)$ will return a value of 1 if n is an odd number. From the **[Menu]** select **Control** followed by **IF ... THEN ... ELSE ... ENDIF**.

Between the words "IF" and "Then" type:

$$\text{Mod}(n,2)=1$$

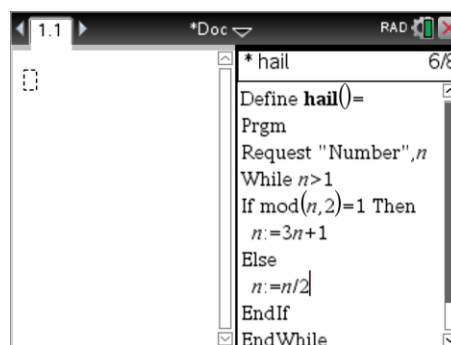
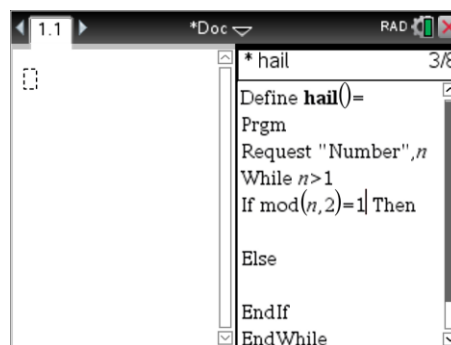
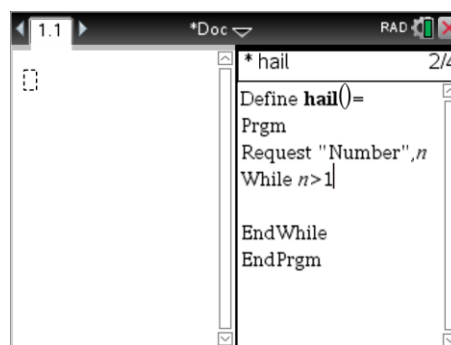
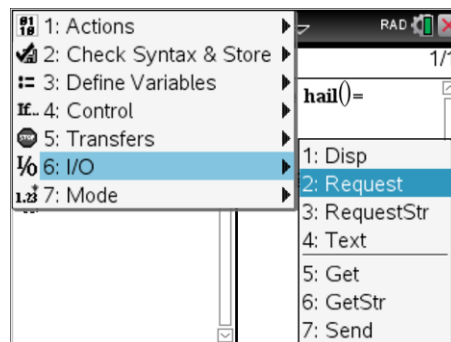
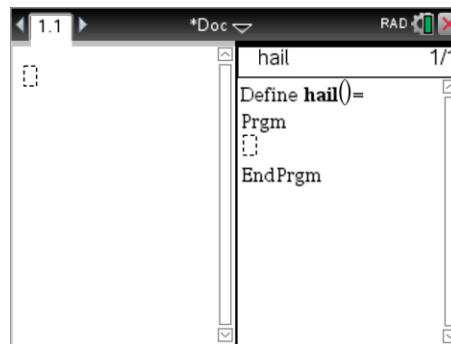
If $\text{Mod}(n,2)=1$ Then the value of n must be multiplied by 3 and one added to the result ... and then stored back in n .

To achieve this result, after the word "Then" , press **[Enter]** to start a new line, then type:

$$n := 3n + 1$$

After the word "Else", press **[Enter]** to start a new line then type:

$$n := n / 2$$



Move the cursor to a line between Endif and Endwhile, if one doesn't exist, place the cursor after Endif and press [Enter]. The program needs to show the progress of the sequence now that n has been changed.

From the I/O menu select **Disp** and then type n .

$Disp\ n$

Once this is done, press **Ctrl + B** to store the program. Your program is now ready. Press **Ctrl + Tab** to shift focus back to the calculator application, away from the programming tool.

Press the [Var] key and select the "Hail" program.

The text Hail() will appear on the screen, press [Enter] to run the program and enter a number, start small so you don't get 'too much hail'.

To review the numbers in the sequence use the up / down and left / right arrows to scroll through the sequence.

```

1.1 *Doc RAD
"haul" stored successful
Prgm
Request "Number",n
While n>1
If mod(n,2)=1 Then
  n:=3·n+1
Else
  n:=n/2
EndIf
Disp n

```

haul() 8/9

Number 45 =1 Then

OK Cancel

```

Prgm
Request "Number",n
While n>1
If mod(n,2)=1 Then
  n:=3·n+1
Else
  n:=n/2
EndIf
Disp n

```

Question: 8.

Use your program to determine how many numbers are in the '45' sequence?

Sequence length = 17

Question: 9.

Use your program to determine how many numbers are in the '47' sequence ... and comment on any short comings of the program.

Sequence length = 105. Sequence is very long, moves too fast to count and takes a long time to count manually, and probably prone to errors.

A counting system can be incorporated into the program to save scrolling. To shift focus back to the program press

Ctrl + Tab.

Insert a line between Prgm and Request. A 'counter' should always be reset at the start of a program to ensure it is not adding on to a previously stored value.

$c := 1$

The counter will start with '1' representing the start number.

Enter a line between Disp and EndWhile, this is where the counter will be increased every time the loop makes another decision, odd or even.

$c := c + 1$

```

1.1 *Doc RAD
5
16
8
4
2
1
Done
* haul 1/10
Prgm
c:=1
Request "Number",n
While n>1
If mod(n,2)=1 Then
  n:=3·n+1
Else
  n:=n/2
EndIf

```

```

1.1 *Doc RAD
5
16
8
4
2
1
Done
* haul 10/11
While n>1
If mod(n,2)=1 Then
  n:=3·n+1
Else
  n:=n/2
EndIf
Disp n
c:=c+1
EndWhile

```

One final instruction, for the user, when the loop has finished the quantity of numbers in the sequence needs to be displayed.

Between EndWhile and EndProgram type the line:

Disp "Seq. Length",c

Save the changes to the program by pressing **Ctrl + B**.

Change focus back to the calculator, **Ctrl + Tab**.

Try some of the numbers from Question 4 to check that your program is working before continuing.

```

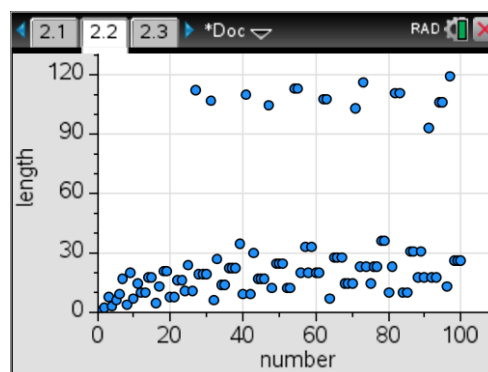
* hail
n:=3:n+1
Else
n:=n/2
EndIf
Disp n
c:=c+1
EndWhile
Disp "Seq. Length",c
EndPrgm

```

Question: 10.

Challenge your friends to see which number(s) between 1 and 100 produce the longest sequence? Record your research and findings.

A graph of all the sequences for starting numbers 1 through to 100 is shown opposite. It is of interesting note that 54 and 55 both produce a sequence length of 113. Similarly, 62 and 63 produce a sequence length of 108. Another pair: 82 and 83 produce a sequence length of 111. The longest sequence is 119 terms produced by the starting number: 97.



Other noteworthy contributions:

(27, 112), (31, 107), (41, 110), (47, 105), (71, 103), (73, 116), (91, 93)

It is also worthy of note that 31, 41, 47, 71, 73, 83, 91 and 97 are all prime. This means in the distinct top group of 16 sequences half of them originate from prime numbers, a solid performance from a relatively small subset of whole numbers between 1 and 100.

Exploration

Mathematicians use patterns and logic to explore. It's your turn to explore the Hail Stone problem. A few starting ideas are presented here, choose one of these or discuss your own idea with your teacher.

Responses to these questions will vary. For calculator programs that will produce some of the findings automatically refer to the companion activity "Hail Stone Inquiry" or have students write their own program if they are well versed in programming.

- Graph sequence length versus starting number and discuss any patterns you find.
 - Numbers between 1 and 100 produce two distinctly different sets of sequences. There are several 'pairs' such as 54 and 55 that produce long sequences (113) but no triplets.
 - Barely distinguishable, is a group of small sequences, becomes more apparent as the start number increases, it is the set of numbers of the form 2^n which of course will contain no upward movement at all.

- Explore hail stone sequences in terms of odds and evens and discuss/explain any patterns you find.

More even numbers appear in the hail stone sequences ... data will show this, also consider an algebraic approach, achievable at a range of levels particularly if CAS is used where expanding has not yet been taught. Any odd number can be expressed in the form: $(2n + 1)$, it is then multiplied by 3 and one is added ... $3(2n + 1) + 1 = 6n + 4$ the result of which is even. So every odd is immediately accompanied by an even number, but even numbers can go to either odd or even. It is this bias that allows the sequence to decrease rather than increase.

- John says “Since 50% of whole numbers are odd, and multiplying by 3 is more ‘powerful’ than dividing by 2, sequences should get bigger not smaller.” Discuss John’s statement, supporting your comments with numerical (or algebraic) evidence.

As above ... odd numbers (by observation) can be seen to always be followed by an even number. Students may choose to use algebra to show this or make written attempts to explain. So the first part of John’s statement is correct, however it doesn’t allow for the bias toward even numbers. Statistically the even numbers occur almost twice as frequently as the odd numbers, therefore producing a division by 2 twice as often (effectively dividing by 4) compared with multiplying by 3, hence the hail stones tend to move downward.

- Prime numbers are used in many areas such as encryption, because of their ‘lack’ of factors. Explore the average length of sequences produced by prime numbers and compare with the sequence length of composite numbers.

The `isPrime` command on the calculator can be used to test larger numbers or simply applied to an entire list. A range of other commands are also available such as ‘countif’ in the spreadsheet if the process needs to be further automated. Students can explore the percentage of ‘large sequences’ that start with a prime number ... bearing in mind the smaller initial percentage of prime numbers amongst the entire set of starting numbers.

- Write your own inquiry brief. (Confirm your brief with your teacher.)

Explore sequence endings ... whilst every sequence ends in 1, what other numbers occur at the end of the sequence? For example ‘2’ will always be included, and once the starting number has exceeded 2, the number 4 will be included. Do all sequences with a starting number greater than 4 include 8 in the final terms of the sequence? This investigation would involve working backwards ... a much more challenging problem to solve.

So we see that it is not long before 16 must be included, with the sequence ending in $2^0, 2^1, 2^2, 2^3, 2^4$... it seems reasonable to expect that eventually 2^5 would have to be included. This however is not true, working backwards show us that 16 can be reached if the preceding term is 5 or 32. In general, whilst terms in the trailing part of the sequence will be of the form 2^n , if $2^n - 1$ is divisible by 3 then 2^n has two different ways of being generated. Have students explore: $\text{mod}(2^n - 1, 3)$.

Example:

Let $n = 4$. $2^{2n} = 256$. $2^{2n} - 1 = 255$, which is divisible by 3. $255 \div 3 = 85$ So the hail stone sequence for the number 85 is: 85, 256, 128, 64, 32, 16, 8, 4, 2, 1. The other way this trailing sequence can be generated is: **512**, 256, 128, 64 ...

In summary, as soon as a sequence arrives at any number of the form 2^n it is guaranteed to return to 1 as it will no longer degenerate into an odd number and therefore no longer increase.