Conjecture

Answers

7 8 9 10 11 12

TI-Nspire | Coding | Student | 60 min

Teacher Notes

This activity is designed to encourage students to see part of the 'bigger picture' in mathematics. Our brains are pre-wired to look for patterns in the world around us. The intellectual discourse between Goldbach and Euler confirming and clarifying each other's observations is prefaced by the comment: "although I cannot prove it". This is the critical message underpinning this activity.

Students understand that one example does not prove a theory; however they often accept multiple examples as 'proof'. On the contrary, students are also accepting of the notion that a single counter example can disprove a theory. What students need to appreciate is the inability to find a counter example does not constitute proof. Our legal system accepts 'beyond reasonable doubt'. A lack of 'certainty' in many aspects of lives allows rational arguments to produce polar opposite belief structures; "I believe in aliens" vs "I don't believe in aliens". Our search for life elsewhere in the universe is really about finding one example that will prove 'aliens' exist, rather than an exhaustive proof of the contrary. Similarly, students undertaking this activity are really seeking to find that one even number that disproves this conjecture.

In the theoretical world of mathematics we seek proof, a compilation of logical conclusions based on a set of axioms. Mathematical proof can often involve algebra, although it is rarely introduced in this way. The notion that a variable can represent 'any number' is extremely important. Coding supports the understanding of a variable and this activity should promote the use of algebra as a tool for generality and proof.

Conjectures

Mathematicians and scientists often develop theories, generally based on observations. They experiment with their theory and then attempt to prove it. If they are not able to prove their theory it is referred to as a conjecture. It is reported that on 7th June in 1742 that Christian Goldbach wrote a letter to Leonhard Euler in which he proposed: "Every integer greater than 2 can be written as the sum of three primes". At the time the number one (1) was considered prime. Correspondence between the two mathematicians continued noting that "Every even integer greater than 2 can be written as the sum of two primes" … "although I cannot prove it."

This exchange is remarkable as mathematicians have still not managed to prove it, so it remains a conjecture and one of the many 'unsolved mathematics problems'. In this investigation you will explore and test, many times over, whether every even number greater than 2 can be written as the sum of two prime numbers.

Author: P. Fox

Question: 1.

Write each of the following even numbers as the sum of exactly two prime numbers, the first two have been done for you.

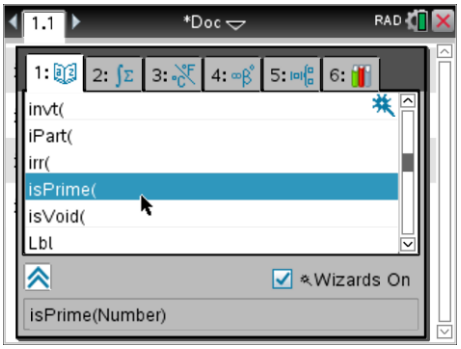
4	6	8	10	12	14	16	18
2 + 2 = 4	3 + 3 = 6	5 + 3 = 8	3 + 7 = 10 5 + 5 = 10	5 + 7 = 12	3 + 11 = 14 7 + 7 = 14	3 + 13 = 16 5 + 11 = 16	5 + 13 = 18 7 + 11 = 18

Instructions

It would take some time just to explore the first 50 even numbers. Programs are a great way to perform such highly repetitive tasks. Fortunately the TI-Nspire calculator includes a very useful command:

'isPrime"

The command can be typed directly or collected from the catalogue.



Question: 2.

Use the isPrime command to test the following numbers: Example: isPrime(97)

Note: The first result has been completed for you.

2	13	29	2003	2011	2013	2017	2019
True	True	True	True	True	False	True	False

Question: 3.

Use your answers to the previous question to help write the following years as the sum of two prime numbers:

Note that other answers are possible, the answers provided here however use the information gleaned from the previous question.

2014	2016	2018	2020	2022	2024	2026	2028
2011 + 3	2003 + 13 2011 + 5	2011 + 7	2003 + 17 2017 + 3	2003 + 19 2011 + 11 2017 + 5	2011 + 13 2017 + 7	2003 + 23	2011 + 17 2017 + 11

Author: P.Fox

Coding a Solution

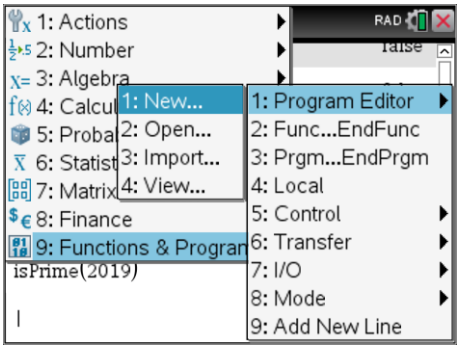
Insert a program application into your document by selecting:

> Functions and Programs > Program Editor > New

Enter the title: **Goldbach** for the name of the program and press [**Enter**].

Initially the page will be split into two with the calculator application on the left and the program on the right.

If you would like to see the pages separate press **Ctrl** + **6**.

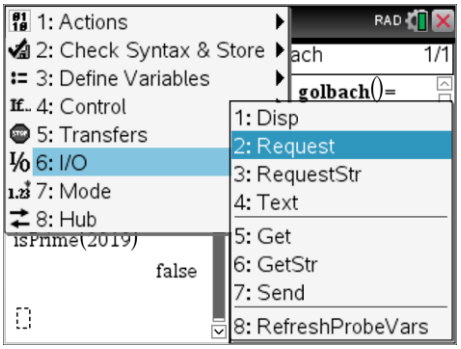


The first instruction in the program will be to request a number from the user.

> I/O > Request

Edit the command as follows:

Request "Enter an even number", n

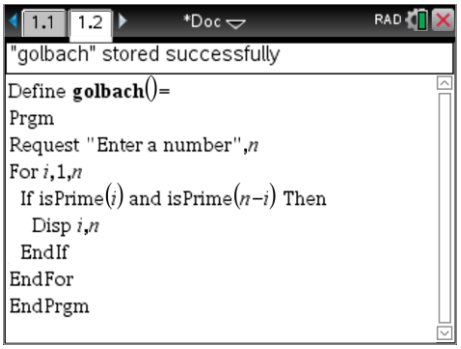


The program will now enter a loop, searching for a pair of prime numbers that add to give 'n', the required number.

> Control > For … EndFor

The FOR loop can start at 1 and end its search at n

For i, 1, n



The program needs to check if the current value of the counter i and the corresponding partner $n-i$ combine as a pair of prime numbers.

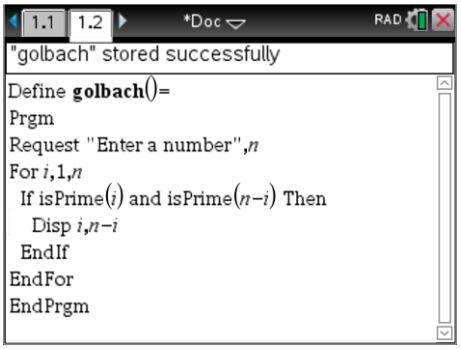
> Control > If … Then … EndIf

Both numbers can be checked at the same time.

If isPrime(ι) **and** isPrime($n-i$) **Then**

If both of these conditions are met then the calculator will display the pair of numbers.

Disp i, n



Press **Ctrl** + **B** to save and compile the program.

Question: 4.

Navigate back to the calculator page. Use the [**VAR**] key to access and run the program. Use your program to find a pair (or pairs) of prime numbers and check your answers to Question 1.

The program will find all answers … students may have missed some of the previous answer options.

Author: P.Fox

Question: 5.

Run the program and check if there are more answers to the years selected in Question 3.

2014	2016	2018	2020	2022	2024	2026	2028
35 solutions	73 solutions	28 solutions	41 solutions	59 solutions	32 solutions	32 solutions	62 solutions

Question: 6.

Explain why changing the **FOR** loop to terminate at n/2 would allow the program to run in approximately half the time while still producing all the prime pairs.

When the program runs through to: *n*, ordered pairs are supplied such as (3, 7), (5, 5) and (7, 3). Terminating the program at n/2 means only (3, 7) and (5, 5) would be returned. The program would run twice as fast as the loop would be executed for only half the numbers.

Question: 7.

Alex changes his **FOR** loop to: **For** *i*, 3, *n*/2, 2 . His program runs even faster as it only checks every second number starting with 3.

a) Explain the logic behind checking every second number, starting with 3.

As 2 is the only even prime number, pairing it with another prime number would mean pairing with an odd number. The sum of an odd number and an even number will produce an odd number. The conjecture refers to even numbers only. By starting at 3 and increasing by 2 {3, 5, 7, 9 …} reduces the quantity of numbers to be checked by 50% therefore increasing the speed of the program.

b) Which number will be missed by Alex's program? Suggest a possible fix or adjustment.

The only number to be missed by Alex's program is 4 since 2 + 2 = 4. The input statement could be changed to "Enter an even number greater than 4". Alternatively, a simple check at the start of the program could be included to see if the user entered '4' … or perhaps an odd number!

These two parts (a) & (b) represent an opportunity for students to use algebra to support their reasoning in regards to 4 being the only number where '2' can be one of the prime numbers in the sum. This supports Alex's reasoning and programming efficiency strategy.

Question: 8.

Which 2 digit even number has the most pairs of prime numbers?

Record the results of your exploration.

10	**12**	**14**	**16**	**18**	**20**	**22**	**24**	**26**	**28**	**30**
2	1	2	2	2	2	3	3	3	2	3
32	**34**	**36**	**38**	**40**	**42**	**44**	**46**	**48**	**50**	**52**
2	4	4	2	3	4	3	4	5	4	3
54	**56**	**58**	**60**	**62**	**64**	**66**	**68**	**70**	**72**	**74**
5	3	4	6	3	5	6	2	5	6	5
76	**78**	**80**	**82**	**84**	**86**	**88**	**90**	**92**	**94**	**96**
5	7	4	5	8	5	4	9	4	5	7
98					Solution Set for 90:					
3		(7, 83)	(11,79)	(17,73)	(19,71)	(23, 67)	(29,61)	(31,59)	(37,53)	(43,47)

Author: P.Fox

Question: 9.

Explain why checking all the even two digit numbers does not prove Goldbach's conjecture.

The conjecture is applied to **all** even numbers, not just two digit numbers. Providing evidence of 3 digit numbers, 4 digit numbers, 5 digit numbers … still does not provide proof that the conjecture will always hold as there are infinitely many numbers … only need to find one that doesn't work.

Ironically the larger 2 digit numbers have more solutions than the smaller 2 digit numbers. The challenge questions (below) provide an opportunity for students to explore whether or not there are any patterns in the number of solutions and the size of the number being tested.

Challenge Problems

The Goldbach program returns all the prime pairs that add to a specified number. If we are trying to prove that the conjecture is true, and we are confident that the calculator is correctly finding pairs of prime numbers, it would be more efficient to simply report how many pairs have been found rather than identify the actual numbers.

Example: The number: 100 contains 6 pairs of prime numbers:
(3, 97), (11, 89), (17, 83), (29, 71), (41, 59) and (47, 53).

Challenge 1:

Change the Goldbach program to report only the quantity of prime pairs.
Include your complete program listing.

Sample Program listing

```
Define Goldbach2()=
Prgm
Request "Enter a number",n                  /No check for even entry
c:=0                                        /Set counter to zero
For i,3,((n)/(2)),2                         /check odd numbers only
    If isPrime(i) and isPrime(n-i) Then     /Both numbers must be prime
        c:=c+1                              /Increase count by one if 'true'
    EndIf
EndFor
Disp "Number of pairs: ",c                  /Display result of count
EndPrgm
```

Challenge 2:

Change your Goldbach program from Challenge 1 to search all even numbers between two specified numbers returning the quantity of prime pairs found for each even number in the specified range. For example, the program output for a search from 100 to 120 would show 100,6; 102, 8; 104, 5; 106, 6; 108, 8 …

The program provides a greater challenge as it involves a nested loop.

Author: P.Fox

```
Define Goldbach3()=
Prgm
Request "Start number",n                    /Note that there is no check that the user
Request "End number",m                      /has entered an even number.
    For p,n,m,2                             /Start at small number (n) through to (m)
        c:=0                                /Reset counter for each number tested
        For i,3,((p)/(2)),2                 /Search current number
            If isPrime(i) and isPrime(p-i) Then   /Both numbers must be prime
                c:=c+1                      /Increase counter when true
            EndIf
        EndFor                              /End of loop for current number
        Disp p,c                            /Display current result
    EndFor                                  /End of loop for numbers being tested.
EndPrgm
```

For more advanced programming options, storing results in a list would allow for a quicker transition for exploring potential patterns in the number of solutions compared with the size of the number(s) being tested. ie: Is it generally true that larger numbers will have more prime pairs? This could also be useful for searching for cyclic patterns.

Students could also be encouraged to search for other patterns, for example, even numbers that end in zero, do they have more solutions than other even numbers?

Other opportunities for students to explore include: Lemoine's conjecture:

"Every large odd number (n > 5) is the sum of a prime and the double of a prime."

Students can explore this conjecture to see if it just a natural extension of Goldbach's conjecture. It should be noted however that this problem needs a little more thought to code at the "IF" statement, otherwise it is essentially the same structure.

Author: P.Fox