

Euclid's Algorithm



Teacher Notes

7 8 9 10 11 12



TI-Nspire™



Activity



Student



120 min

Euclid's Algorithm

Teacher Notes:



A PowerPoint slide show is provided with this activity as an introductory presentation for students to watch and help them understand how the algorithm works. The slides use 'lengths' to help explain why the algorithm works.

Following the 'lengths' examples, numbers are included to see how Euclid's algorithm translates to working with numbers



The calculator contains a command for the "Highest Common Factor" or "Greatest Common Divisor", it is however good for students to understand how the 'magic' happens.

The GCD command only compares 2 numbers, in the later stages of this activity, students extend their program to 3 numbers and then an entire list!



TI-Codes Lessons:

Unit 1 – Skill Builder 1



Unit 4 – Skill Builder 1

Commands:

- input
- while
- if
- else
- int (number types)
- print

Highest Common Factors

The Highest Common Factor (HCF) or Greatest Common Divisor (GCD) of two numbers is useful for many reasons. The process is valuable when working with fractions, solving packaging problems, developing traffic light sequences and encrypting content for digital communications. Developed more than 2000 years ago, Euclid's algorithm is still the most efficient process used to determine the Highest Common Factor of two numbers.



Euclid's Algorithm:

- LINE #1: IF $A = 0$ THEN $GCD(A,B) = B$ since $GCD(0,B) = B$
- LINE #2: IF $B = 0$ THEN $GCD(A,B) = A$ since $GCD(A,0) = A$
- LINE #3: $A = B \times Q + R$... where Q is the quotient and R is the remainder
- LINE #4: $GCD(B,R) = GCD(A,B)$, now find $GCD(B,R)$

This algorithm will make more sense when some numbers are used for A and B . Suppose we want to find the highest common factor of (A) 1260 and (B) 385. As neither $A = 0$ or $B = 0$ we progress to LINE #3.

$$1260 = 385 \times 3 + 105 \quad [\text{We say that 105 is the remainder when 1260 is divided by 385}]$$

According to LINE #4 of Euclid's algorithm: $\text{GCD}(1260,385) = \text{GCD}(385,105)$

We apply the algorithm again. Since $385 \neq 0$ and $105 \neq 0$ we proceed to LINE #3.

$$385 = 105 \times 3 + 70 \quad [\text{We say that 70 is the remainder when 385 is divided by 105}]$$

According to LINE #4 of Euclid's algorithm: $\text{GCD}(1260,385) = \text{GCD}(385,105) = \text{GCD}(105,70)$.

We apply the algorithm again. Since $105 \neq 0$ and $70 \neq 0$, we proceed to LINE #3

$$105 = 70 \times 1 + 35 \quad [\text{We can say that 35 is the remainder when 105 is divided by 70}]$$

We are getting close! According to LINE #4 of Euclid's algorithm: $\text{GCD}(1260,385) = \dots = \text{GCD}(70,35)$

Applying the algorithm one more time, as $70 \neq 0$ and $35 \neq 0$, we proceed to LINE #3.

$$70 = 2 \times 35 + 0. \quad [\text{This time the remainder is 0!}]$$

Now we can apply LINE #1 or LINE #2 since we have $\text{GCD}(35,0) = 35$.

Our conclusion is that the Highest Common Factor or Greatest Common Divisor of 1260 and 385 is 35.

Question: 1.

Use Euclid's algorithm to identify the highest common factor of: 3850 and 3234.

Answer: 154

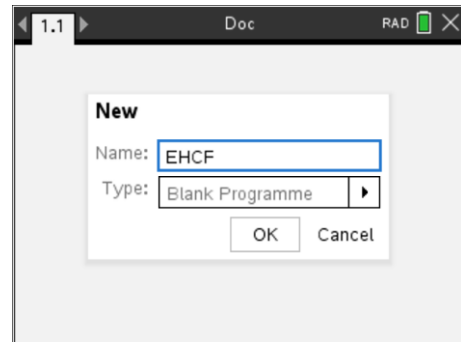
Writing a Program

Instructions:

Start a new document; insert a new Python program.

Add Python > New

Call the program: EHCF



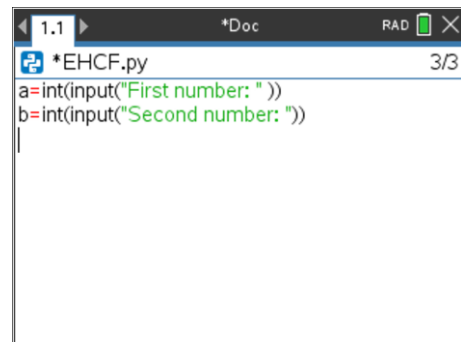
Insert two prompts for the integers 'a' and 'b'. The input type (integer) can be typed directly or entered via the menu.

Menu > Built-ins > Type

The input command can be typed directly or entered via the menu:

Menu > Built-ins > I/O

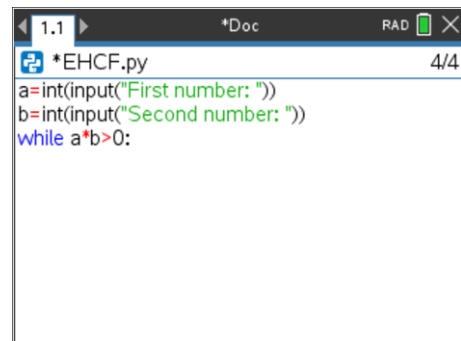
Include a text prompt for the user.



Euclid's algorithm ceases when either $a = 0$ or $b = 0$, an easy way to check this is: $a \times b = 0$. The "null factor law" states that if the product of two numbers is zero, then one or both of the numbers must be zero.

The algorithm should continue to run while $a \times b \neq 0$.

Menu > Built-ins > Control > while..



Modular arithmetic returns the remainder when $a \div b$ (where $a > b$) so an `if ...else` statement can be used to process Line #3 of Euclid's algorithm.

Menu > Built-ins > Control > if ... else

Enter the condition, then press Tab to navigate to the instruction block and use the % sign for modular arithmetic, then Tab to the second block to complete the instruction block.

```

1.1 *Doc RAD 8/9
*EHCF.py
b=int(input("Second number: "))
while a*b>0:
  if a>b:
    a=a%b
  else:
    b=b%a

```

That's the entire algorithm! The only thing remaining is to display the results. You could display as:

```
print(a,b)
```

Alternatively, use an `if ... else` option to only display only the highest common factor rather than both a and b. (Shown opposite.)

```

1.1 *Doc RAD 12/14
*EHCF.py
b=int(input("Second number: "))
while a*b>0:
  if a>b:
    a=a%b
  else:
    b=b%a
if a==0:
  print("HCF: ",b)
else:
  print("HCF: ",a)

```

Question: 2.

Determine the highest common factor of: 1914 and 7293 (by hand) using Euclid's algorithm and use your results to check the program.

Answer: 33

Question: 3.

Test your program on some smaller numbers where you know the highest common factor. Record your test results.

Answer: Answers will vary depending on what numbers student chose to explore and test.

Question: 4.

The **Number** menu in the Calculator Application contains a command to determine the highest common factor of **two** numbers. Edit your program to find the highest common factor of three numbers.

Example: EGCD(a,b,c)

Test and evaluate your program.

Answer: There are various ways students may edit their program to achieve the desired result. Students should also consider efficiency.

Teacher Notes: Sample Program

A simple addition to the original program is shown here. Following the original While loop, an IF statement is used test which variable, a or b, is equal to zero, the corresponding variable is then replaced with c.

If a = 0 then c is stored in a, otherwise c is stored in b.

```

1.1 1.2 *Doc RAD 19/22
EHCF.py
if a==0:
  a=c
else:
  b=c
while a*b>0:
  if a>b:
    a=a%b
  else:
    b=b%a
if a==0:
  print("HCF: ",b)

```

The original While loop can then be pasted in its entirety.

Conceptually, copying and pasting the original loop leads students to an understanding that the original loop is repeated and therefore extending the program to a list of numbers involves an over-arching loop.

Students should provide a table of numbers that they have tested. Students should also think about how they generate the numbers to be tested. For example:

$$a = 29 \times 35 = 1015 \quad b = 41 \times 35 = 1435 \quad c = 97 \times 35 = 3395$$

Generating the numbers as products with primes ensures the highest common factor of the three numbers is 35 in the above scenario.

Question: 5.

Edit your program to test for the highest common factor of an entire list of numbers.

Note: The program can be defined as `egcd(data)` where `data` is a list of numbers: `{#, #, # ...}`. The `len` command can be used to determine the dimensions (quantity of numbers) entered into the list.

Answer:

Sample program opposite:

- While loop works for data entry with the "esc" key as the criteria to finish data entry
- The quantity of entries in the 'data' variable relates to the number of times the HCF loop needs to operate.
- Original while loop is also included with successive entries called from the 'data' variable.

For very large data sets it would be worthwhile sorting a list in descending order.

```
print("Enter your numbers, ESC to finish")
data=[]
a=None
while a != "esc":
    ♦♦a=input("Number: ")
    ♦♦if a!="esc":
        ♦♦♦♦a=int(a)
        ♦♦♦♦data.append(a)
d=int(len(data))
a=data[0]
print(a)
for n in range(1,d):
    ♦♦b=data[n]
    ♦♦print(a,b)
    ♦♦while a*b>0:
        ♦♦♦♦if a>b:
            ♦♦♦♦♦♦a=a%b
        ♦♦♦♦else:
            ♦♦♦♦♦♦b=b%a
    ♦♦if a==0:
        ♦♦♦♦a=b
    if a==0:
        ♦♦print("HCF: ",b)
    else:
        ♦♦print("HCF: ",a)
```

Investigation

The prime factorisation of a number can be used to efficiently find the highest common factor of any two or more numbers. Use your program to find the highest common factor for each list of numbers (below). Write the original numbers and the highest common factor in terms of their prime factorisation. Try some of your own lists, then write a description of how you can use the prime factorisation to determine the highest common factor of any two or more numbers.

List 1: 1260, 1410, 2040, 4290 & 9570

List 2: 220, 1400, 1700, 30940 & 154700

List 3: 2964, 3588, 8892, 10764 & 409032

List 4: 399, 441, 1911, 3381, 5733 & 835107