Factor Count

Student Activity

7 8 **9** 10 11 12

TI-Nspire™ Activity Student 120 min

TI-Codes Lessons:

Unit 1 – Skill Builder 1
⇩
Unit 4 – Skill Builder 1

Commands:

- input
- for (range)
- if
- print
- int (number types)
- [ ] (create a list)
- Append (add elements to a list)
- len (length of a list)
- % (modular arithmetic)

Finding and Counting Factors

There are many ways to determine the quantity of factors for a specified number. The most common method is to test the divisibility for all applicable numbers. For example, suppose we want to determine the quantity of factors for 18. We can determine the quotient and remainder for all the numbers from 1 to 18.

Table 1A – Finding the factors of 18

Divisor	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Quotient	18	9	6	4	3	3	2	2	2
Remainder	0	0	0	2	3	0	4	2	0

Table 1B – Finding the factors of 18

Divisor	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Quotient	1	1	1	1	1	1	1	1	1
Remainder	8	7	6	5	4	3	2	1	0

Our conclusion is that 18 has six factors since there are six occasions whereby the remainder is equal to zero.

This divisibility check for all numbers is exhaustive. You may have ideas about how this process can be made more efficient, however, this method will provide a basis for an algorithm on which to write a simple program to count the quantity of factors for a given number. You can make the necessary improvements and checks once your initial program is complete and functioning.

Question: 1.

Write a description of the steps required to determine the quantity of factors for any whole number: n.

Author: P. Fox

Instructions:

Start a new document and insert a calculator application.

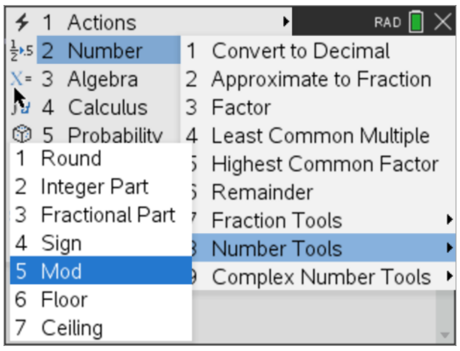
Locate the **mod** command using: **Number** > **Number Tools** > **Mod**

Determine the result of the following calculations:

Mod(18,6)

Mod(18,5)

Mod(18,12)



Question: 2.

Based on your experimentation, what value does the MOD command return?

Question: 3.

If MOD(*a*, *b*) = 0, what does this say about the relationship between *a* and *b*?

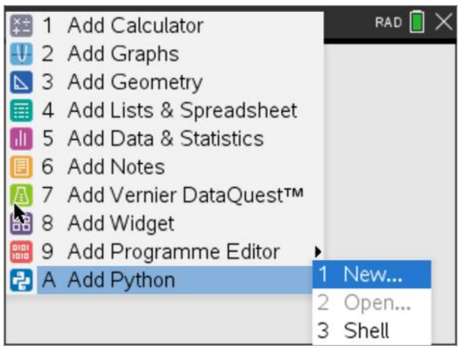
Writing a Program

Create a new Python program by pressing:

ctrl + doc▾ , **Add Python** > **New...**

Call the program: FactorCount

Note that 'FactorCount' is one word as program names cannot contain spaces.



If the programming application is launched on the same page as the Calculator Application. The page-layout in the document menu can be used to give each application its own page.
Short-cut: [**Ctrl**] + [**6**]. Page 1.1 = Calculator Application. Page 1.2 = Program Application.

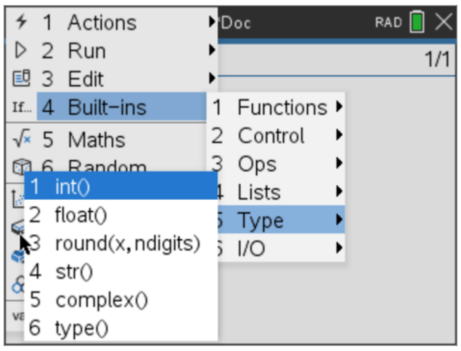
The first task is to request a number from the program user and store it as a variable. Type in:

n =

Python input defaults to text, so the next step is to restrict the input to a whole number (integer). The int() command is located in the "**type**" menu, alternatively it can be typed in directly from the keyboard:

Press:

menu > **Built-ins** > **Type** > **int()**

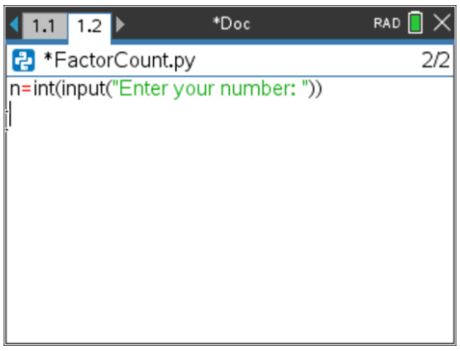


The next step is to enter the “input” command:

menu > **Built-ins** > **I/O** > **input()**

Finish the instruction by adding a text prompt.

When this command is executed, the user’s numerical input will be stored in a variable “n”



```
n=int(input("Enter your number: "))
```

- Quotation marks: **“ ”** can be entered by pressing [**Ctrl**] + [**x**] (multiplication sign)
- Colour is added automatically to help locate the various parts of the syntax.

We could just count factors, however, it is easy record and store them in a list which also helps with checking the results.

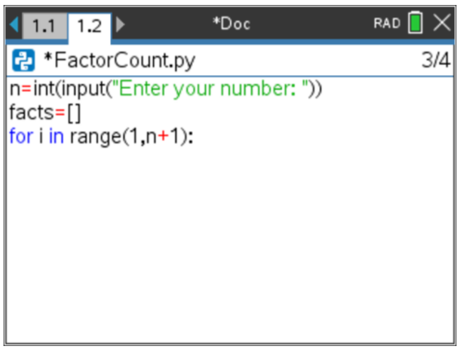
Factors = [ ]

This creates an empty list called factors.

A “FOR” loop can be used to check for factors. We use a FOR loop because we can pre-determine the quantity of iterations the loop must perform.

menu > **Built-ins** > **Control** > **For index in Range(***size***)**

Python loop execution ceases when the counter reaches *size*, therefore the counter (*size*) needs to be one more than n.



```
n=int(input("Enter your number: "))
facts=[]
for i in range(1,n+1):
```

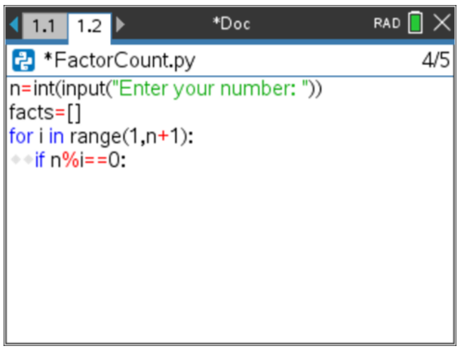
An **IF** statement will be used to check if the user’s number has a factor each time the program executes the loop.

The IF command can be selected by:

menu > **Built-ins** > **Control** > **if**

The % operation in Python is for modular arithmetic.

The percentage sign can be obtained from the punctuation fly-out menu.



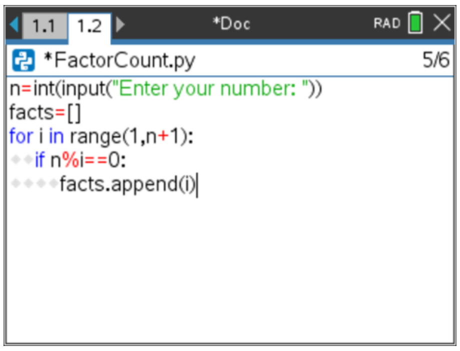
```
n=int(input("Enter your number: "))
facts=[]
for i in range(1,n+1):
  if n%i==0:
```

The ‘append’ command adds the latest factor to the current list of factors. This command can be typed in directly or accessed from the variable menu (facts) followed by:

menu > **Built-ins** > **Lists** > **.append()**

The value to be added to the list, given that the division has not generated any remainder, is “*i*”, the loop counter.

That’s all for the algorithmic part of the program! The next step is to display the results. Start by deleting the indentions, the end of the “IF” condition and the For loop.



```
n=int(input("Enter your number: "))
facts=[]
for i in range(1,n+1):
  if n%i==0:
    facts.append(i)
```

Author: P. Fox

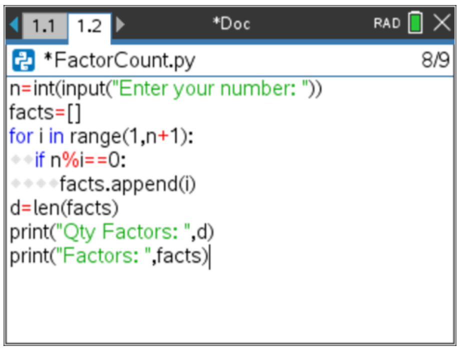
The list ("facts") contains all the factors, len(facts) therefore returns the size of the list. This quantity can be stored in 'd'.

Now the quantity of factors (d) and the actual factors can be printed to the screen.

Save the program and launch it by pressing [**Ctrl**] + [**R**]. A new Python shell will be created and the program name automatically pasted. Start by checking the factor count for 18.

The table at the start of this activity identifies 6 factors, compare this with the output from your program.

To check another number, press [**Ctrl**] + [**R**]



```
n=int(input("Enter your number: "))
facts=[]
for i in range(1,n+1):
  if n%i==0:
    facts.append(i)
d=len(facts)
print("Qty Factors: ",d)
print("Factors: ",facts)
```

Question: 4.

Determine the quantity of factors for each of the following numbers:

a. 24

b. 36

c. 37

d. 144

Check each of your answers by writing down all the factors.

Question: 5.

Determine the quantity of factors for each of the following numbers. Identify a specific characteristic about the quantity of factors and use this to classify the numbers into two groups, explain your classification.

29, 84, 104, 87, 22, 37, 101, 97, 45, 43, 133, 153, 173, 107

Question: 6.

Determine the quantity of factors for each of the following numbers. Identify a specific characteristic about the quantity of factors and use this to classify the numbers into two groups, explain your classification.

28, 30, 90, 45, 50, 60, 120, 72, 25, 49, 81, 144, 441, 82, 24, 720.

Question: 7.

The FactorCount program works, but it could be more efficient. Use a stop watch to time how long the program takes to count the number of factors for: 100,000; 200,000 and 300,000. Use these times to predict how long the program will take to count the factors for 500,000. Test your answer!
If you are satisfied with your prediction, how long would it take to find factors for the following number:

2,140,324,650,240,744,961,264,423,072,839,333,563,008,614,715,144,755,017,797,754,920,881,418,023,447,140,136,643,345,519,095,804,679,610,992,851,872,470,914,587,687,396,261,921,557,363,047,454,770,520,805,119,056,493,106,687,691,590,019,759,405,693,457,452,230,589,325,976,697,471,681,738,069,364,894,699,871,578,494,975,937,497,937 [250 digits!]

Note: This number is associated with RSA Encryption.

Investigation

Why are factors important? To answer this question, consider the opposite situation, the *absence* of factors. Billions of dollars are moved around electronically every day, to do this securely, the electronic transfers must be encrypted. The most common encryption method (RSA) is built around very large prime numbers, numbers with an *absence* of factors! All encryption methods are essentially built on numbers, so being able to 'assemble' and 'disassemble' numbers is extremely important.

Your task is to find a rule that determines the quantity of factors for any whole number, given the prime factorisation of that number.

A few clues are provided along the way to help you on your factor sleuth journey. Document your search findings and conclusions using the clues and your program to help expedite your investigation.

Clue 1:

Determine the prime factorisation and corresponding quantity of factors for the following numbers: 36; 100; 441; 3025 & 48841.

Clue 2:

Determine the prime factorisation and corresponding quantity of factors for the following numbers: 24; 250; 1029; 6655 and 198911.

Clue 3:

Based on the first two clues, generate some numbers that you believe have exactly 8 factors. Test your numbers and comment on the results.

Clue 4:

Determine the prime factorisation and corresponding quantity of factors for the following numbers: 2000; 64827; 107811; 668168 and 1585615607. [Note: For this last number you will need a fast algorithm!]

Clue 5:

Create some numbers of the form: $m^2 \times n^5$ where m and n are both prime. Determine the quantity of factors for each of your numbers. [Note: You may want to choose relatively small prime numbers for m and n.]

Continue the exploration, tabulate your results and record your thoughts, hypotheses, tests and reflections as you go. Documenting findings is an important part of the investigative process. Detectives may have many suspects in their initial investigations, however as more clues surface they develop hypotheses. Detectives test each hypothesis, review what they already know or go in search of more clues. Some investigations end up as Cold Cases, however it is critical that detailed documentation of all aspects of their investigation are retained in the event the investigation is re-opened. Some crimes remain unsolved despite having significant suspects, in mathematics these are often called 'conjectures', a theory that seems to work but has never been proven.