



A First Program – Calculating the Drake Equation

Dr. Frank Drake conducted the first search for radio signals from extraterrestrial civilizations in 1960. This established SETI (Search for Extraterrestrial Intelligence), a new area of scientific inquiry. In order to estimate the number of civilizations that may exist in our galaxy that we may be able to communicate with, he developed what is now called the

Drake equation: $N = R \cdot p \cdot n \cdot f \cdot i \cdot c \cdot L$

The Drake equation accounts for a number of different factors. The values used for some of these are the result of scientific study, while others are only the result of an “intelligent guess.”

The factors consist of:

- R , the average rate of star creation per year in our galaxy;
- p , the percentage of those stars that have planets;
- n , the average number of planets that can potentially support life for each star with planets;
- f , the percentage of those planets that actually go on to develop life;
- i , the percentage of those planets that go on to develop intelligent life;
- c , the percentage of those that have the technology to communicate with us; and
- L , the expected lifetime of civilizations (the period that they can communicate).

The Drake equation is simply the multiplication of all these factors, giving N , the estimated number of detectable civilizations there are at any given time,

Drake Equation Factor Values		Estimated Values
Rate of star creation	R	7 [†]
Percentage of stars with planets	p	40%
Average number of planets that can potentially support life for each star with planets	n	(no consensus)
Percentage of those that go on to develop life	f	13%
Percentage of those that go on to intelligent develop life	i	(no consensus)
Percentage of those willing and able to communicate	c	(no consensus)
Expected lifetime of civilizations	L	(no consensus)
† Estimate of NASA and the European Space Agency		

The table shows those parameters in the Drake equation that have some consensus as to their correct value.

The Problem

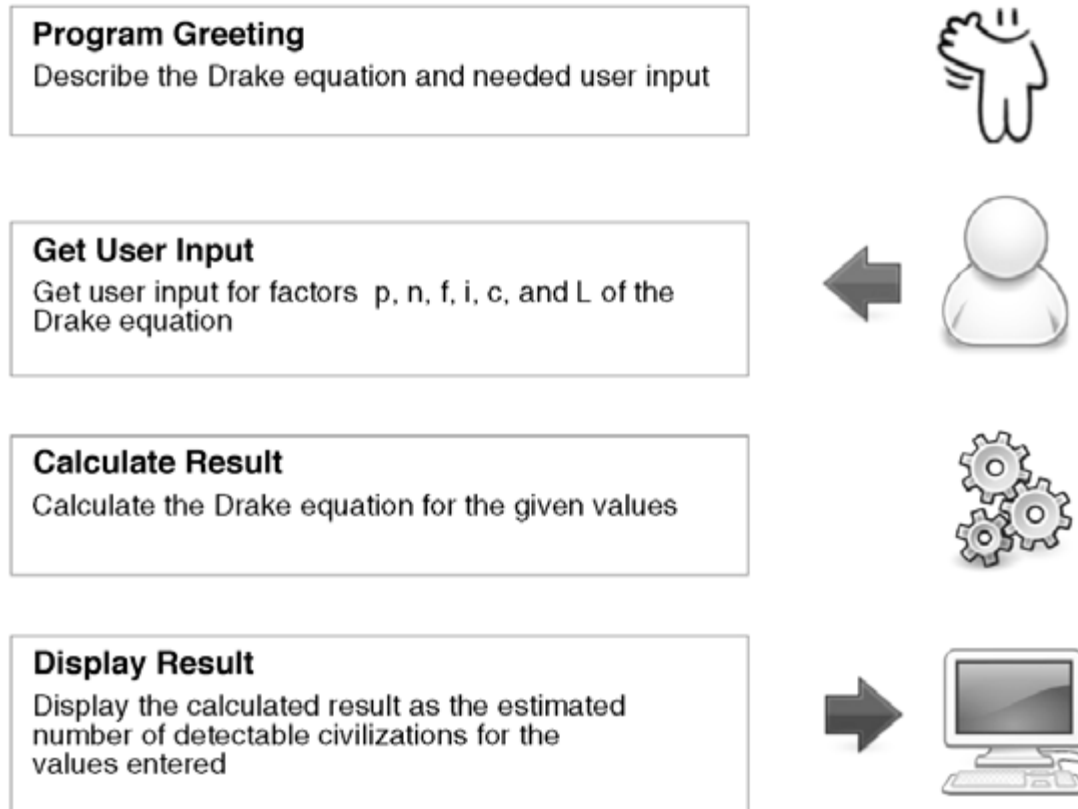
The value of 7 for R , the rate of star creation, is the least disputed value in the Drake equation today. Given the uncertainty of the remaining factors, you are to develop a program that allows a user to enter their own estimated values for the remaining six factors (p , n , f , i , c , and L) and displays the calculated result.

Problem Analysis

This problem is very straightforward. We only need to understand the equation provided.

Program Design

The program design for this problem is also straightforward. The data to be represented consist of numerical values, with the Drake equation as the algorithm. The overall steps of the program are depicted below.



Program Implementation

The implementation of this program is fairly simple. The only programming elements needed are input, assignment, and print, along with the use of arithmetic operators. The implementation is given below.

Your task is to create a new Python project called SET_yourLastName and enter the code below - exactly as written.

```

1 # SETI Program
2 #
3 # The Drake equation, developed by Frank Drake in the 1960s, attempts to
4 # estimate how many extraterrestrial civilizations, N, may exist in our
5 # galaxy at any given time that we might come in contact with,
6 #
7 #     N = R * p * n * f * i * c * L
8 #
9 # where,
10 #
11 #     R ... estimated rate of star creation in our galaxy
12 #     p ... estimated percent of stars that have planets
13 #     n ... estimated average number of planets that can potentially support
14 #         life for each star with planets
15 #     f ... estimated percent of those planets that actually go on to develop life
16 #     i ... estimated percent of those planets go on to develop intelligent life
17 #     c ... estimated percent of those that are willing and able to communicate
18 #     L ... estimated expected lifetime of such civilizations
19 #
20 # Given that the value for R, 7 per year, is the least disputed of the values,
21 # the user will be prompted to enter estimated values for the remaining six
22 # factors. The estimated number of civilizations that may be detected in our
23 # galaxy will then be displayed.
24
25 # display program welcome
26 print('Welcome to the SETI program')
27 print('This program will allow you to enter specific values related to')
28 print('the likelihood of finding intelligent life in our galaxy. All')
29 print('percentages should be entered as integer values, e.g., 40 and not .40')
30 print()
31
32 # get user input
33 p = int(input('What percentage of stars do you think have planets?: '))
34 n = int(input('How many planets per star do you think can support life?: '))
35 f = int(input('What percentage do you think actually develop life?: '))
36 i = int(input('What percentage of those do you think have intelligent life?: '))
37 c = int(input('What percentage of those do you think can communicate with us?: '))
38 L = int(input('Number of years you think civilizations last?: '))
39
40 # calculate result
41 num_detectable_civilizations = 7 * (p/100) * n * (f/100) * (i/100) * (c/100) * L
42
43 # display result
44 print()
45 print('Based on the values entered ...')
46 print('there are an estimated', round(num_detectable_civilizations),
47       'potentially detectable civilizations in our galaxy')

```

Notes:

First, note the program lines beginning with the hash sign, #. In Python, this symbol is used to denote a comment statement. A **comment statement** contains information for persons reading the program. Comment statements are ignored during program execution—they have no effect on the program results. In this program, the initial series of comment statements (lines 1–23) explain the Drake equation and provide a brief summary of the purpose of the program.

Comment statements are also used in the program to denote the beginning of each program section (lines 25, 32, 40, and 43). These **section headers** provide a broad outline of the program following the program design.

The program welcome section (lines 25–30) contains a series of print instructions displaying output to the screen. Each begins with the word `print` followed by a matching pair of parentheses. Within the parentheses are the items to be displayed. In this case, each contains a particular string of characters. The final “empty” print, `print()` on line 30 (and line 44), does not display anything. It simply causes the screen cursor to move down to the next line, therefore creating a skipped line in the screen output. (Later we will see another way of creating the same result.)

The following section (lines 32–38) contains the instructions for requesting the input from the user. Previously, we saw the input function used for inputting a name entered by the user. In that case, the instruction was of the form,

```
name = input('What is your name?:')
```

In this program, there is added syntax,

```
p = int(input('What percentage of stars do you think have planets?:'))
```

The `input` function always returns what the user enters as a string of characters. This is appropriate when a person’s name is being entered. However, in this program, numbers are being entered, not names. Thus, in, the following,

```
What percentage of stars do you think have planets?: 40
```

40 is to be read as single number and not as the characters '4' and '0'. The addition of the `int (. . .)` syntax around the input instruction accomplishes this.

On line 41 the Drake equation is calculated and stored in variable `num_detectable_civilizations`. Note that since some of the input values were meant to be percentages (`p`, `f`, `i`, and `c`), those values in the equation are each divided by 100. Finally, lines 44–47 display the results.

Program Testing

First, you must test the program execution. If you Run, the program, you should see the following. If you do not, or get an error message, go back and edit/revise your code, as needed, until the program executes correctly.

Case 1	100%	10	100%	100%	100%	10,000			
Extreme Case 1	100%	12	100%	100%	100%	10,000			

Based on your results, mark “passed” in the Evaluation column, if the program results are confirmed by the calculator results.