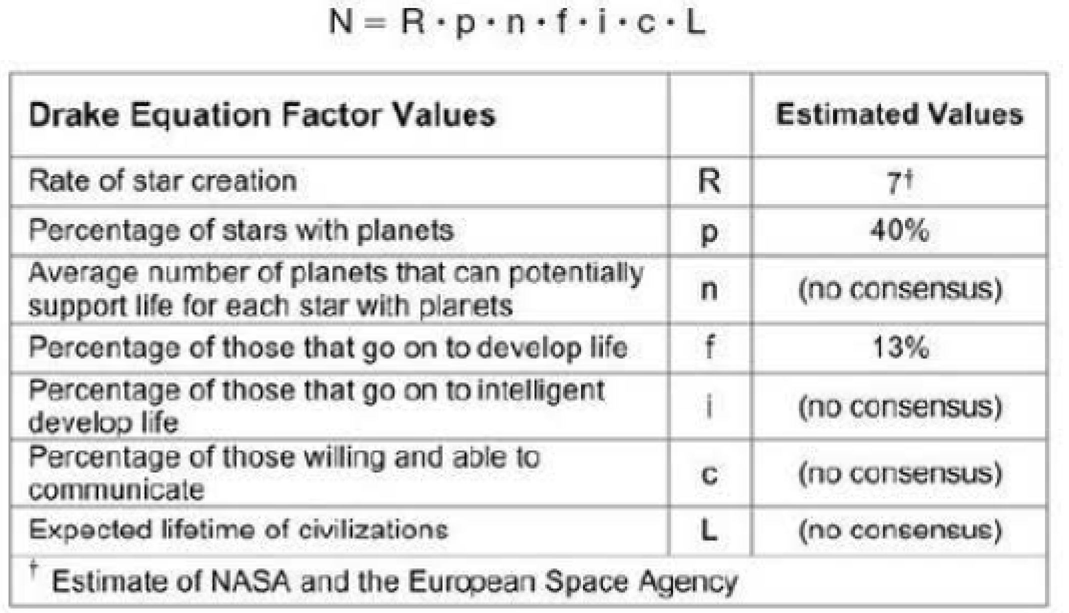
Drake Equation

Professor Caleb Fowler
1/10/24

Problem.

The Drake equation is named after Dr. Frank Drake. He was the first radio astronomer to look for intelligent life in the universe. He developed this equation to answer the question, "How many potential alien civilizations are there in the universe? ". You are going to write a program to calculate the Drake equation. You will declare and initialize the variables in the formula and display the output. Use the formula in figure 1 for your work. The N is the number of potential alien civilizations.

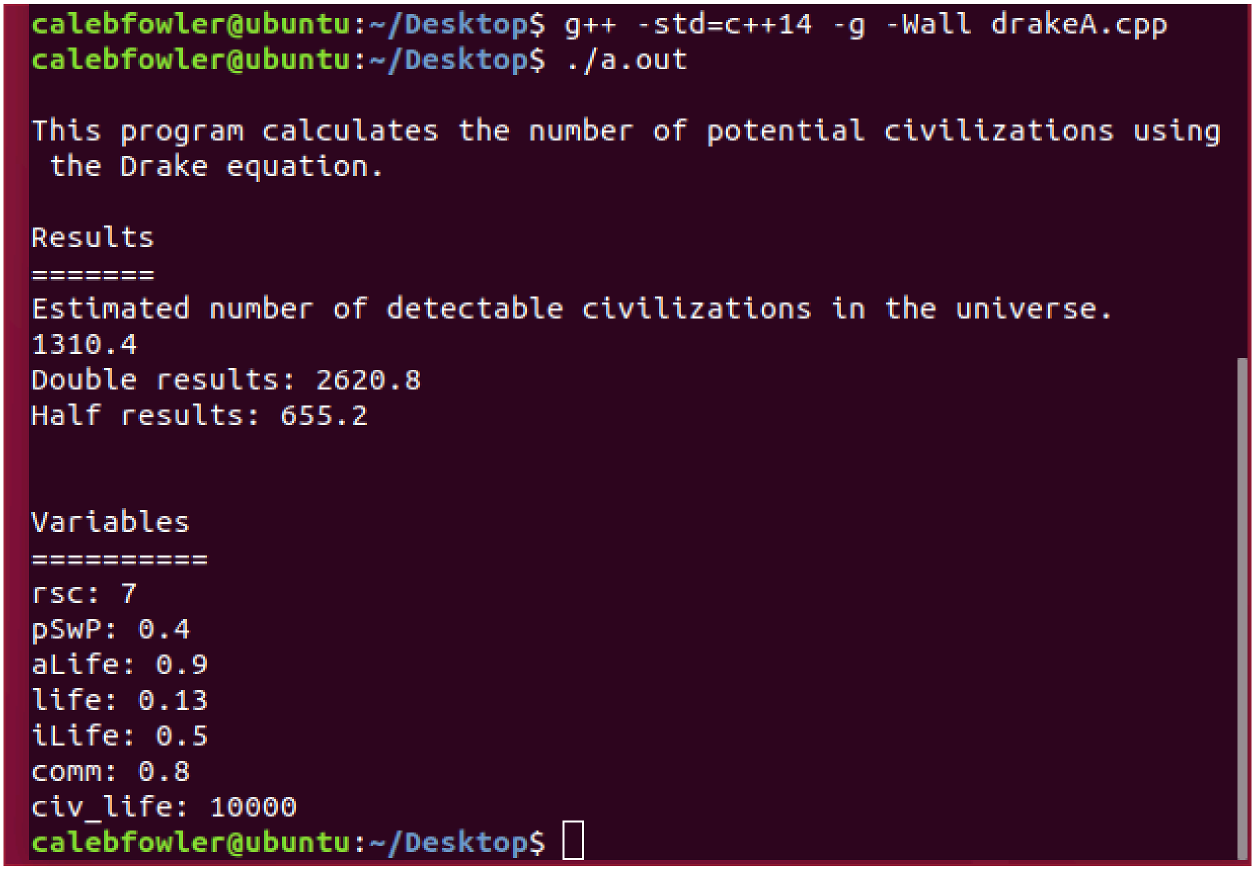


$$N = R \cdot p \cdot n \cdot f \cdot i \cdot c \cdot L$$

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

Figure 1: The Drake equation with commonly accepted values. No consensus means the scientific community has not formulated a value for this parameter. You can use any value you wish for ANY variable. However, if you use the values I used in Figure 3 you should get an answer similar to mine. That would verify your logic is working correctly.



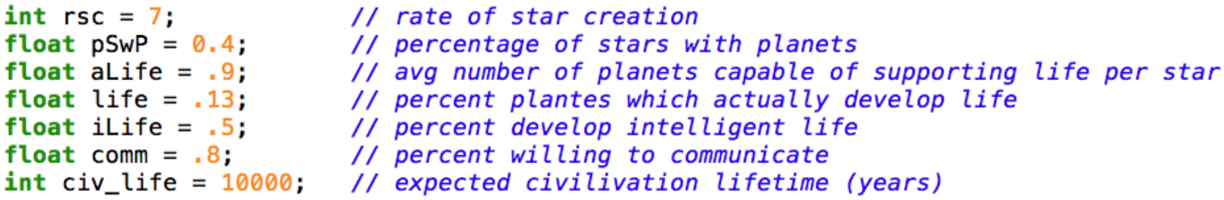
```
calebfowler@ubuntu:~/Desktop$ g++ -std=c++14 -g -Wall drakeA.cpp
calebfowler@ubuntu:~/Desktop$ ./a.out

This program calculates the number of potential civilizations using
 the Drake equation.

Results
=======
Estimated number of detectable civilizations in the universe.
1310.4
Double results: 2620.8
Half results: 655.2


Variables
==========
rsc: 7
pSwP: 0.4
aLife: 0.9
life: 0.13
iLife: 0.5
comm: 0.8
civ_life: 10000
calebfowler@ubuntu:~/Desktop$ 
```

Figure 2: Sample program run. This output may not be complete or in corformance with the specifications. However, it is correct given the listed values.



```
int rsc = 7;               // rate of star creation
float pSwP = 0.4;          // percentage of stars with planets
float aLife = .9;          // avg number of planets capable of supporting life per star
float life = .13;          // percent plantes which actually develop life
float iLife = .5;          // percent develop intelligent life
float comm = .8;           // percent willing to communicate
int civ_life = 10000;      // expected civilivation lifetime (years)
```

Figure 3: Sample values for the various variables.

Requirements.

These are the requirements for the assignment. This means they are general and apply to the entire assignment, rather than one specific part. Not every assignment will have a requirements section.

- Come up with your own variable names. Do NOT use the single letter names from the formula - they are poor variable names. You really don't need variable names larger than 15 characters.

- Hard-code the data into the program. You do not need to worry about prompting the client for additional input.
- Correctly compile your program.
- Successfully execute your program and generate correct results.
- Check the plagiarism score to make sure it's low enough (Close to green). If your score is not green, you need to examine to plagiarism report and find out why. Discuss your findings and conclusions in a comment in the Canvas assignment.
- Use white-space and comments to make your code more readable.
- Program activities are split into logical 'chunks' or paragraphs. I'm expecting paragraphs for input, processing (if any), and output operations.
- Include a **Source File Header**, like you used in homework 1. Do not include a specification with it like you did in that assignment.
- Include a **Commented Sample Run** for this assignment like you did for homework 1. Again, do not include a specification with it.
- Include a **ProgramGreeting** for this assignment like you did for homework 1.

Specifications.

These are the actual requirements the client wants you to implement in your program.

// Specification C1 - Variable Declaration
Create variables to hold your data and calculation. Rename the variables in the formula in Figure 1 to something more understandable. I'm looking for you to use a formula with variables that have values assigned to them.

// Specification C2 - Variable Initialization
Initialize your variables to appropriate values as well as declare them. Assign the integer variable Rate of Star Creation a value of 7.If you declared and initialized in the same step, put the specification comment C1 on the line below the specification comment C2. Specification comments should always have a line to themselves.

// Specification C3 - Constant Variables
Use the proper syntax as well as the const keyword to create appropriate constant variables.

// Specification B1 - Calculation
Put this comment above the code which actually performs the calculation. Do not put the calculation in your cout statement(s), have it store the value in a variable called **etCiv**.

// Specification B2 - Drake Equation Heading
Add a heading to your equation's output which looks like this:

{ blank line }

```
CHANCE OF INTELLIGENT LIFE
==========================
{ Your output goes here.}
```

Format and display the number of potential alien civilizations to **3** decimal places. Use cout for this. Do not be tempted to use the printf() command (ever) as it is not part of the C++ Standard Library.

// Specification B3 - Double and Half Output and Headings
Display on the console your calculation results doubled and reduced by half. This is called sensitivity analysis. The heading should look something like:

```
{ blank line }
SENSITIVITY ANALYSIS
===================
{ Your output goes here.}
```

// Specification A1 - Variable Table
Create a 2 column table and display all the variable names and their values. When you see table in the specifications, you should think something like this as output:

```
|-----------------|-------------------|
|   Column        |     Headers       |
|-----------------|-------------------|
| Row values      | as needed         |
|-----------------|-------------------|
```

// Specification A2 - Output Table
Create a 2 column table and display “**Drake Equation Value**” and the value of **etCiv**.

// Specification A3 - Sensitivity Analysis Table
Create a 2 column table displaying the results of your sensitivity analysis.