This coding challenge was written by Mikayla Scott under the C2 Pipeline Innovation and Curiosity Center through Wayne State University. This challenge has students work to transpose a sheet of music, Mary Had a Little Lamb, into a code for the computer to play. The challenge prompt is listed below. For more information on the STEM Challenges produced by C2 Pipeline visit https://c2pipeline.wayne.edu/stem-lab. The YouTube video that accompanies this lesson can be found at https://www.youtube.com/watch?v=EHIrHHNKxXY.

+-

“Transpose the song Mary Had a Little Lamb using the Ruby programming language. Download the Sonic Pi compiler and code the song. Your program should utilize the use of the play and sleep functions within the program.”

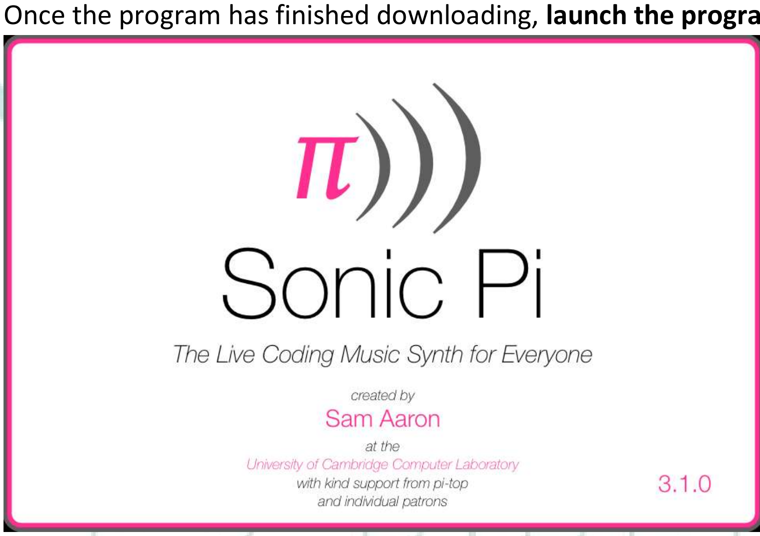
+-

Instructions:

1. Go to https://sonic-pi.net/ and **download the compiler** onto your computer. There are three different options to choose from when you are downloading the software, Windows, MacOS, and Raspberry Pi. Choose your computer and then proceed to download the software.

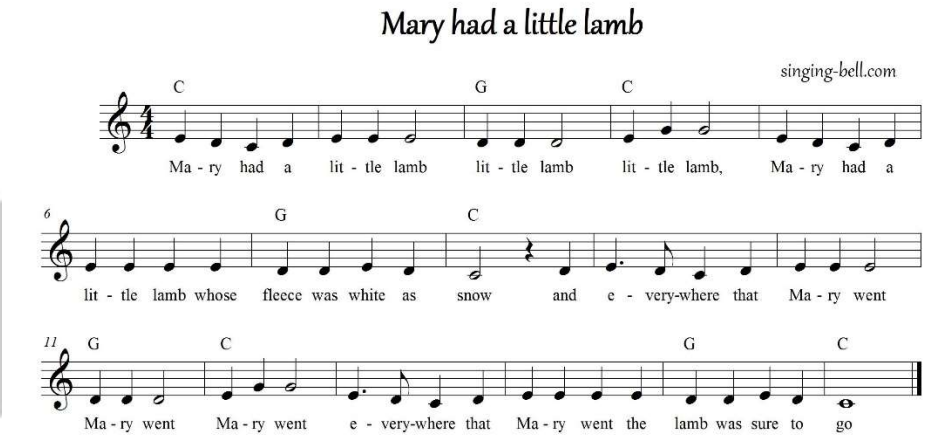
   TIP: If you have a choice between the portable app and the installer, consider the following: the portable app is for flash drives, and the installer will be put onto your computers hard drive.

2. Once the program has finished downloading, **launch the program**.

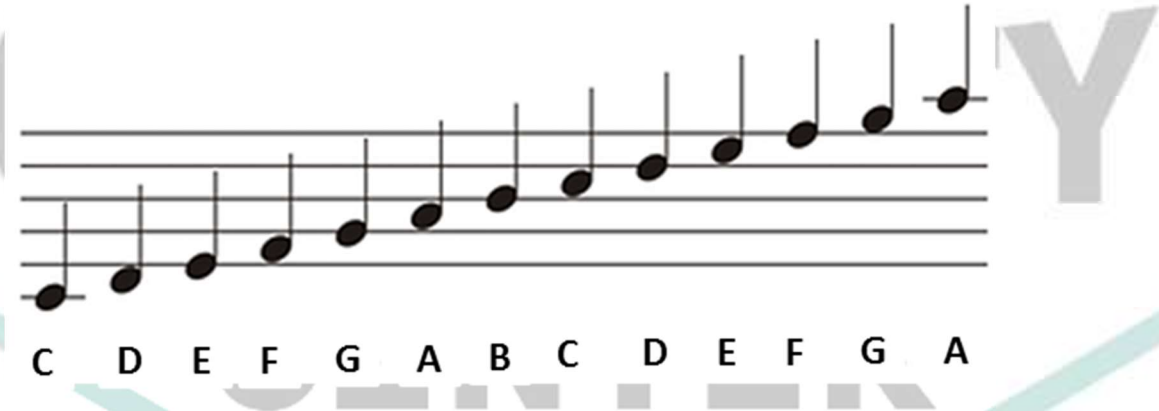


3. While the program is launching go to Google.com/images and **find some sheet music to transpose**, it can be any song. For this coding challenge, the song that was used is Mary Had a Little Lamb. The image used can be seen below.

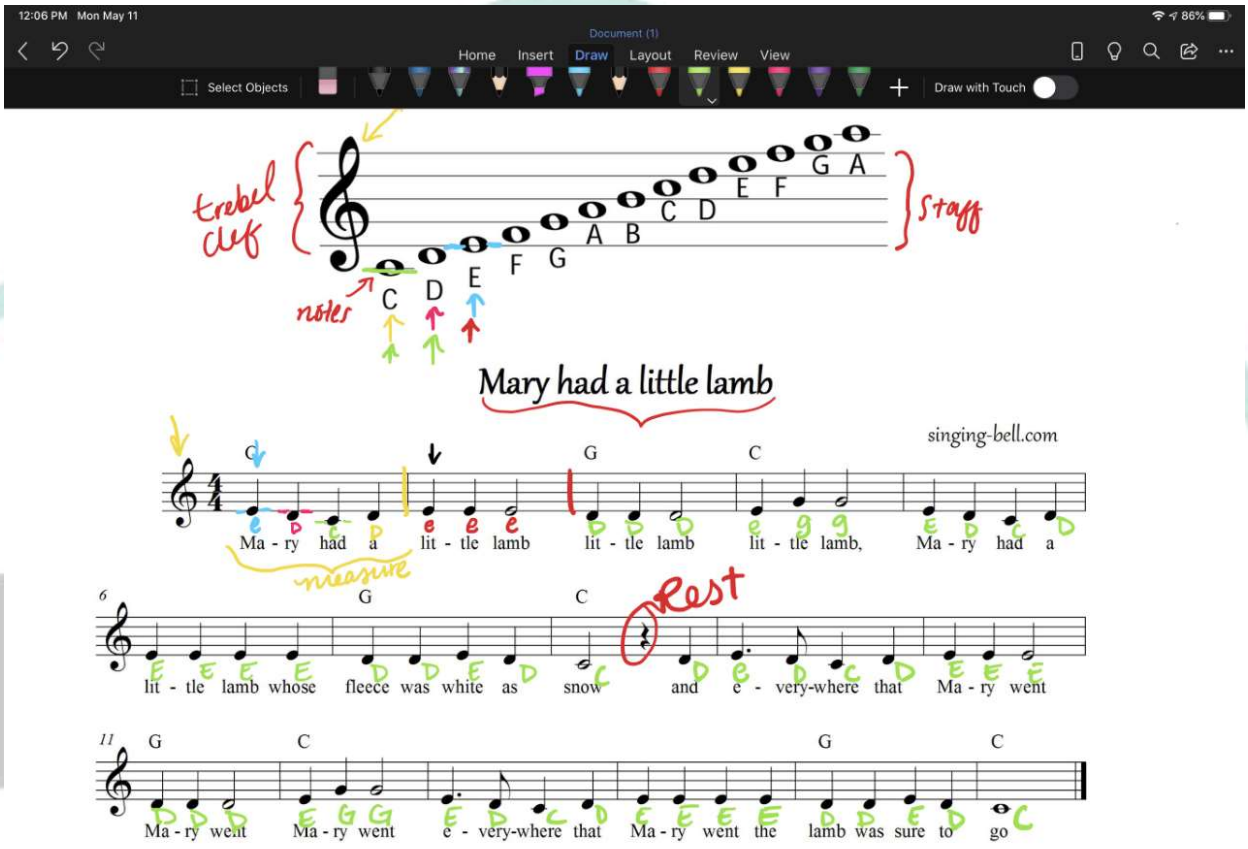
   TIP: Make sure that the song that you choose has a treble clef, and is in 4/4 time. These are the first two things on the musical staff. 



4. While on google find a **musical scale with the name of notes written underneath of it.** This will come in handy for those who have never read music before. This is the one used in the video.



5. Now take your song and **write the name of each note underneath** each note on the sheet of music that you found to tran. See the image below.
   TIP: Print out your document or write the names of the notes on a sheet of paper to make the coding process easier in the long run!



---

6. Once the names of the notes have been written under each of the notes in the song, then work to **write the length of the notes.**

   TIP: The length of the notes must be in decimals, the length of the notes is the parameter for the sleep/rest function in Sonic Pi.

   For a more detailed breakdown on how to get the length values of the notes watch the YouTube video for this challenge.

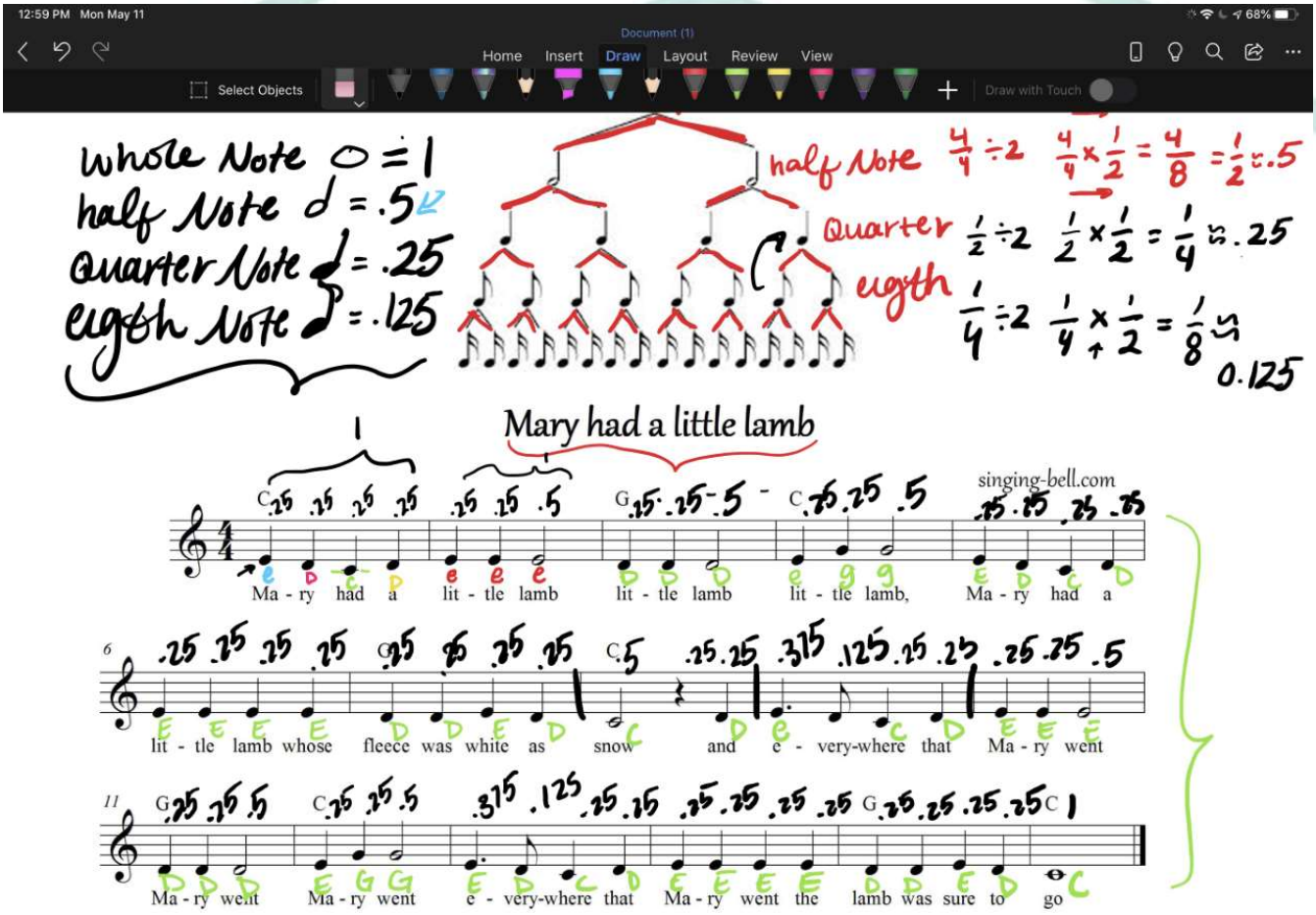
   Below you will see the note lengths that were produced by doing calculations on them.

Whole Note		1
Half Note		0.5
Quarter Note		0.25

CURIOSITY CENTER
C2 PIPELINE

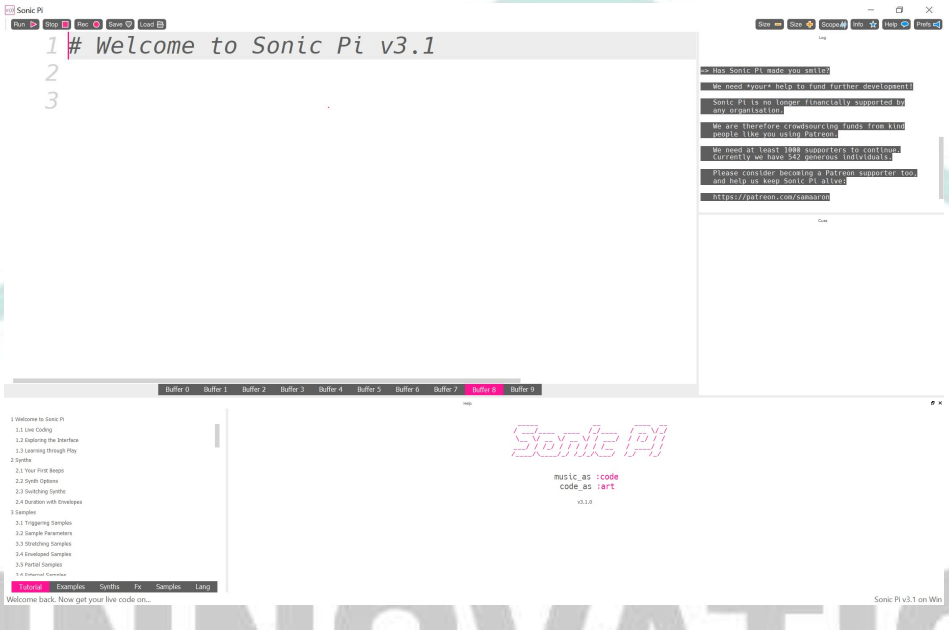
Eighth Note  0.125

Now write the length of the notes on the sheet of the music. See the image below.



7. Your sonic Pi program should look like this:
   TIP: comments in Ruby are denoted by a #, remember that comments don't show up in the initialization of the program, they are only there to help the programmer follow along with what the code is doing.

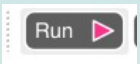
CENTER
C2 PIPELINE



8. Now **find a midi table** online by Google image searching Midi table.
   TIP: The Midi Table is used to convert the names of the notes that we found earlier into numbers to input into the play function

Note	Octave										
	-2	-1	0	1	2	3	4	5	6	7	8
C	0	12	24	36	48	60	72	84	96	108	120
C#	1	13	25	37	49	61	73	85	97	109	121
D	2	14	26	38	50	62	74	86	98	110	122
D#	3	15	27	39	51	63	75	87	99	111	123
E	4	16	28	40	52	64	76	88	100	112	124
F	5	17	29	41	53	65	77	89	101	113	125
F#	6	18	30	42	54	66	78	90	102	114	126
G	7	19	31	43	55	67	79	91	103	115	127
G#	8	20	32	44	56	68	80	92	104	116	---
A	9	21	33	45	57	69	81	93	105	117	---
A#	10	22	34	46	58	70	82	94	106	118	---
B	11	23	35	47	59	71	83	95	107	119	---

YOU WILL BE STARTING IN OCTAVE 3!

9. **Now start to code your program**, going note by note put the number that you acquired from the midi table into the play function. Directly following each play function there should be a sleep function that has the length that we calculated for each note. See the first measure below:
   TIP: Use comments to keep track of measures while you code, after you finish a measure you should run your code to make sure that it sounds correct! 

INNOVATION CENTER
C2 PIPELINE

```
#measure 1
play 64
sleep 0.25
play 62
sleep 0.25
play 60
sleep 0.25
play 62
sleep 0.25
```

10. **Keep repeating step 9 until you get to the end of the song.**
11. Once the song is written you can change the instrument by using the Use_synth : function and you can change the tempo of the song by using the use_bpm function.
    TIP: Write these at the beginning of the program.

```
use_bpm 30
use_synth :piano
```

12. **Run the program!**

FOR MORE INFORMATION

For more information please visit https://c2pipeline.wayne.edu/stem-lab. The full .txt file of this program can be accessed from here as well!

Happy coding, as always!

-Mikayla

INNOVATION & CURIOSITY CENTER
C2 PIPELINE