GRAPH PAPER PROGRAMMING[1]

VOCABULARY

Algorithm - A list of steps that you can follow to finish a task

Programming Language - a formal constructed language designed to communicate instructions to a machine, particularly a computer.

Program - An algorithm that has been coded into a programming language that can be run by a machine

INTRODUCTION TO GRAPH PAPER PROGRAMMING

In this activity, you are going to guide a hypothetical a machine (a robotic arm) that can make drawings. Robots operate off of "instructions," specific sets of things that they have been preprogrammed to do. In order to accomplish a task, a robot needs to have a series of instructions (sometimes called an algorithm) coded into a language (programming language) that the robot can understand and run.

To get more familiar with the concept of an algorithm, it is helpful to have something to compare it to. For this exercise, we will introduce a programming language made of lines and arrows.

Please, watch this instructional video first.

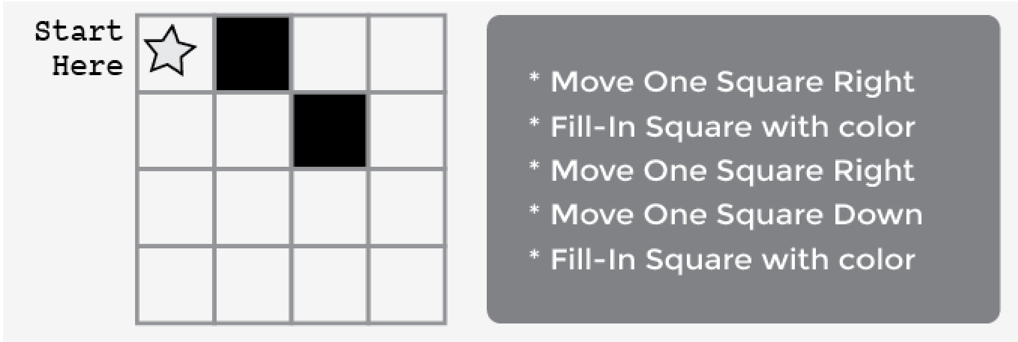
https://www.youtube.com/watch?v=Yy1zbkfRtIg&feature=youtu.be&list=PL2DhNKNdmOtobJjiTYvpBDZ0xzhXRj11N

To illustrate this exercise, we will use sheets of 4x4 graph paper. Starting at the upper left-hand corner, we'll guide our robot with simple instructions. Those instructions include:

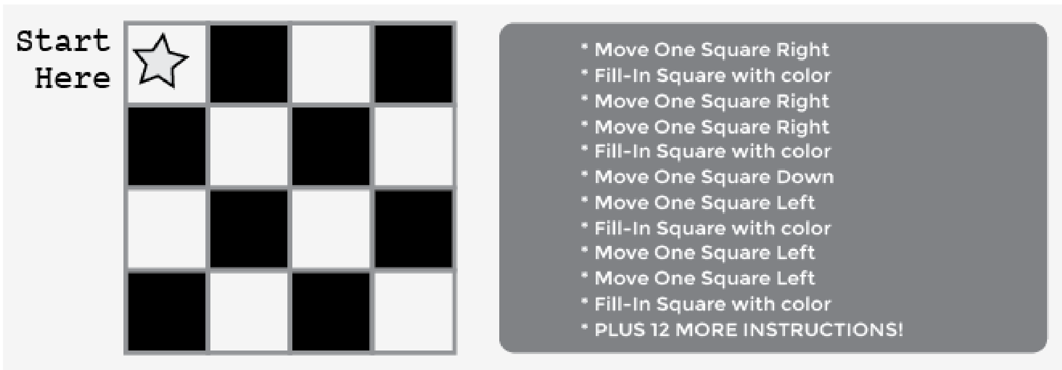
• Move One Square Right	• Move One Square Down
• Move One Square Left	• Fill-In Square with color
• Move One Square Up	

For example, here is how we would write an algorithm to describe our robot has to follow in order to color their blank grid so that it looks like the following image:

[1] Based in the lesson from Code.org : https://code.org/curriculum/course2/1/Teacher

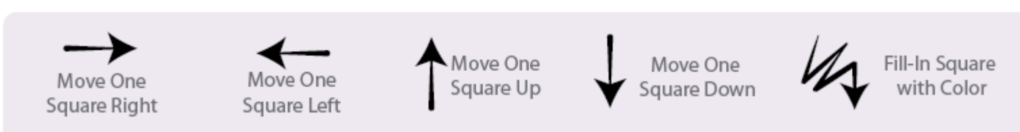


That's simple enough, but it would take a lot of writing to provide instructions for a square like this:



In the examples above, the algorithms are written in English language (more or less). For a robot, reading an algorithm written in English might be not clear, English can be ambiguous. Instead we need a more formal language. We need to give the instructions to our robot in a language that he understands.

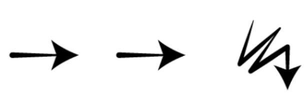
With one little substitution, we can do this much more easily! Instead of having to write out an entire phrase for each instruction, we can use arrows. The following would be the programming language we can use to program our drawing machine.



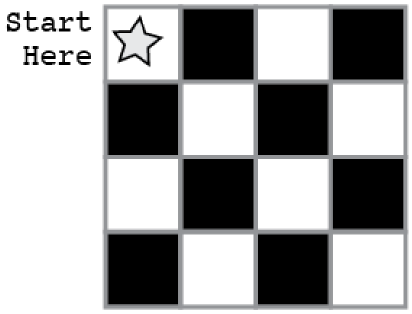
In this instance, the arrow symbols are the instructions that we can use to create our "program". Thus, we can codify our "algorithms" (the words) into a "program". For example:

"Move one square right, Move one square right, Fill-in square with color"

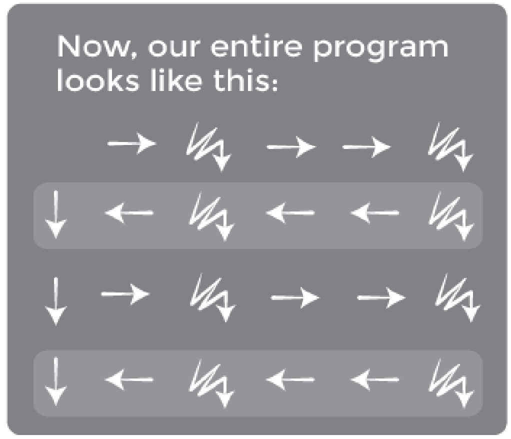
would correspond to the program:



Using our arrow based programming language, we can redo the algorithm from the previous image as follows.

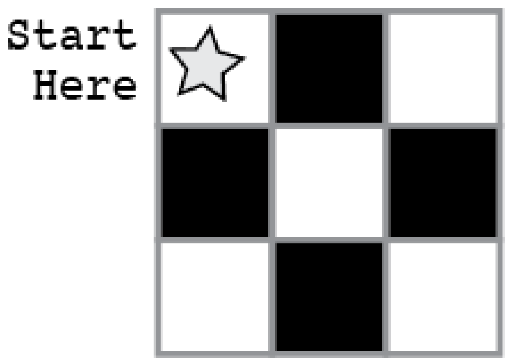


Follow along with your finger and see if you can figure out how to get this image from the program to the right.



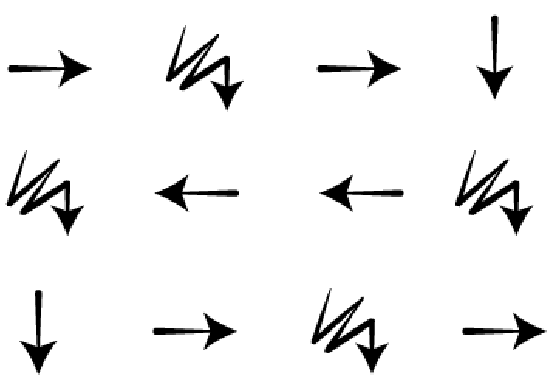
ONE MORE SIMPLE EXAMPLE

What instructions would you need for your robot to complete the following drawing?

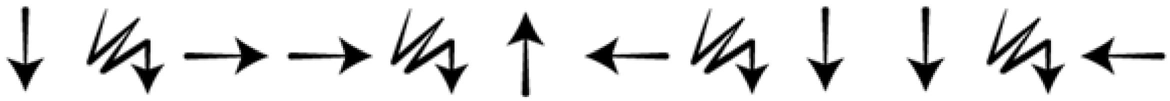


A sample algorithm: “Move Right, Fill-In Square, Move Right, Move Down, Fill-In Square, Move Left, Move Left, Fill-In Square, Move Down, Move Right, Fill-In Square, Move Right”

Or in our graphical language:

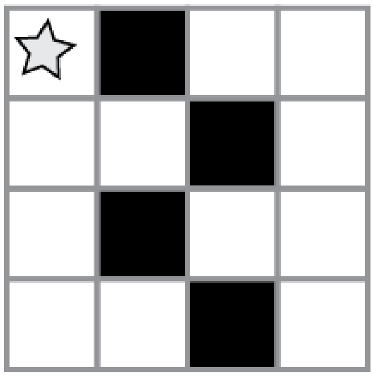


Can we do it in a different way? How about the following program, would it work?



How about the instructions for the next figure?

Start Here



Step 1	2	3	4	5
Step 6	7	8	9	10
Step 11	12	13	14	15
Step 16	17	18	19	20
Step 21	22	23	24	25

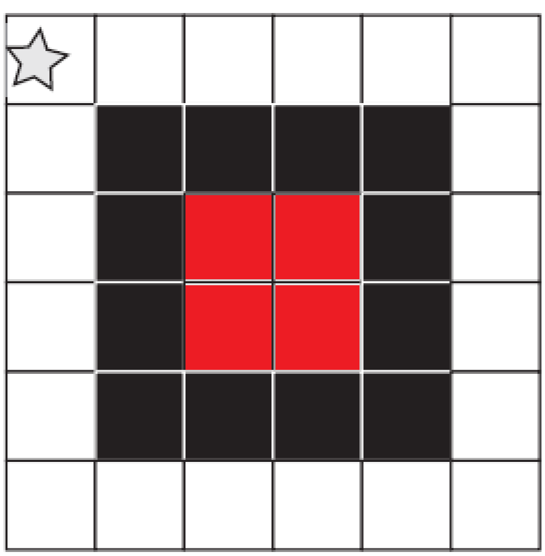
EXTENDING THE PROGAMMING LANGUAGE

Let's add one more instruction to our programming language. This instruction would change the color of the pencil used by the robot.

 — Change to Next Color

Let's assume that the robot has only two colors, so with the instruction above switch between the two colors. If the current pen color is black, what would be the program to complete the following drawing, use the steps grid after the drawing?

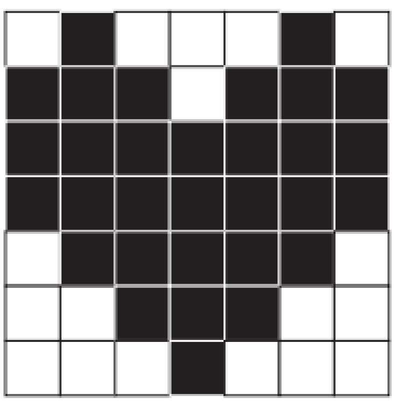
Start Here



Step 1	2	3	4	5	6	7	8	9	10
Step 11	12	13	14	15	16	17	18	19	20
Step 21	22	23	24	25	26	27	28	29	30
Step 31	32	33	34	35	36	37	38	39	40
Step 41	42	43	44	45	46	47	48	49	50

EXTENDING THE PROGAMMING LANGUAGE II (ADDING FUNCTIONS)

Let's look at one more exercise. The algorithm presented here might not be the best (the most efficient) but it does the job.



Algorithm:

"Step forward, fill-in, step forward, step forward, step forward, fill-in, step forward, next line.
Back, back, back, back, back, back.
Fill-in, Step forward, fill-in, step forward, fill-in, step forward, step forward, fill-in, Step forward, fill-in, Step forward, fill-in, next line.
Back, back, back, back, back, back."

First of all, we are using other English words, for humans there is no problem to understand it. However, for our robot this would be a problem.

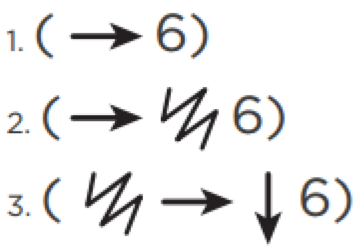
Now, can you identify or look for things that repeat often? Certainly one of the most helpful things to combine is the **"Back, back, back, back, back, back."** How could we transform this sequence into a single symbol? Any suggestions? You probably have several ideas. You might have thought of

$$\leftarrow 6$$

Are there any other combinations that can be made with the same idea? What about skipping three squares in series? Coloring three squares in series? Coloring seven squares in series?

If you have some ideas, you just discovered **functions**! You have created a simple representation for a complex grouping of actions. That is exactly what functions are meant to do. How about the number? It has a name also. That number is called a **parameter**. In the case of the example above, the parameter lets the function know how many times to move backward.

Look at these "functions." What do you suppose they do?



Answers:

1. Move forward six spaces
2. Color 6 blocks in a row
3. Color a diagonal line

Are there any other combinations that could be helpful? Armed with this new method, you can now handle hardest images. Do these symbols help the process go more quickly?

Using the functions above, create a program that would draw the previews image (the heart).

One more try. How about a program for the following drawing.