

Problems, Procedures, Programs, Patterns

In lab we learned the Tractor Control Language and used that language to write programs to solve graphical problems. We also discussed and evaluated programs, applying criteria including length, clarity, efficiency, and the elusive quality of elegance.

We learned about using abbreviations to stand for longer strings of commands. In C++, an abbreviation is called a *function*. You define an abbreviation by saying a single letter stands for a string of commands. You use that single letter as a short-hand way to tell the machine to perform that string.

Functions do two things. They make code shorter although not necessarily more efficient. More importantly, abbreviations make code clearer by representing complex patterns of action with a single name. This technique of creating new vocabulary words and using them is extremely important. Therefore several of these problems require this technique.

As we saw in lab, you can define several different abbreviations *and* you can also use abbreviations within the definition of other abbreviations.

Furthermore, you can use numbers to repeat blocks of code. (see next page for details). By combining basic instructions, abbreviations, repetition, and blank spaces, you can express complex solutions in fairly clearly.

Using the Tractor Simulator

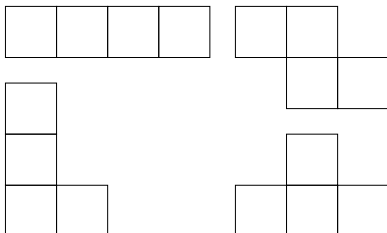
The Tractor Simulator we used in class can be found on the course website

<http://www.cs.tufts.edu/comp/11/labs/01>

There are four buttons to execute code. The first button runs the program. The second button runs the program and also reports several useful numbers about the program and its operation. Try it to see what it tells you. The other buttons present the action in other formats.

Exercise 1-1: Tetris Shapes

These four shapes appear in the computer game Tetris:



For each of these four shapes:

- Write a program to draw it in the orientation shown
- The program must use two or more functions
- In at least two of the shapes, one function must include another function in its definition
- What visual pattern does each function represent?
- What is the minimum number of retraces possible?

(Note: The component squares in these pictures are each one tractor unit long. When drawn in the simulator, these shapes do not look as square. Don't worry.)

Exercise 1-2: Programs and Pictures

Computer scientists do more than devise procedures. They analyze the world of problems and the world of programs. And they compare those two worlds to see how to match up programs and problems. For this exercise you will explore some connections between these two worlds.

Consider programs that are one character long. There are only two of them. There is the program R and there is the program F. Each of those programs creates a picture. The first creates a blank picture. The second creates a single vertical line. Got it? For these problems, ignore abbreviations and consider only programs that consist of F and R commands.

- Now, consider programs that are two characters long.
 - How many different 2-letter programs are there?
 - List all those programs.
 - For each of the programs, what picture does it draw?
 - How many *different* pictures are there?

Note: The same shape in different places are the same, but the same shape at different rotations are not the same.
- Now, consider programs that are three letters long.
 - How many different 3-letter programs are there?
 - List all those programs.
 - For each of the programs, what picture does it draw?
 - How many *different* pictures are there?
- Now, consider programs that are four letters long.
 - Answer the same four questions
- Now, consider programs that are five letters long.
 - Answer the same four questions
- Is the following statement true or false:

"Every shape you can draw with a four-letter program can also be drawn with a five-letter program"

Give a clear explanation or a formal proof.

Can you generalize this statement?

have left out the spaces and written:

```
35S: :
```

which looks like you want 35 of something. Which is confusing, because you really get 15.

Abbreviations and Repeats

As the previous examples show, you are allowed to mix abbreviations and repeats in any combination. An abbreviation can refer to a piece of code that contains a repeat, and you can repeat things that contain abbreviations. Also, you can repeat things that contain repeats.

Use Spaces for Clarity

As the previous example shows, spaces help clarify the structure of your code. The computer does not care if you use spaces. You, though, will find your programs easier to read if you put spaces in to separate abbreviation definitions from each other and to separate sections of the program.