



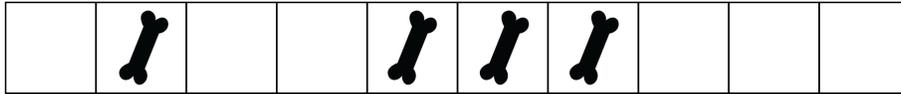
Problem of the Week

Problem C and Solution

Dog Bones

Problem

Elbashir wrote a computer program to control Scruffy the dog as he moves along a row of ten squares, some of which contain a bone, as shown.



Elbashir wrote functions that allow Scruffy to move left or right a given number of squares, pick up a bone, or put down a bone. However, some care needs to be taken when using the functions. Scruffy can hold only one bone at a time, so cannot pick up a bone if he is already holding one. Similarly, he cannot put down a bone if he isn't holding one. Trying to do either of these actions will result in an error and cause the program to stop.

When the program starts, Scruffy is in the leftmost square and is not holding a bone.

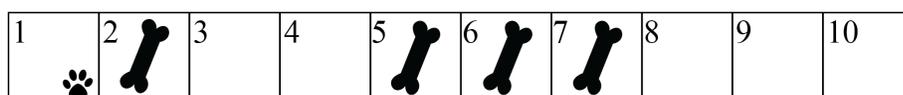
- (a) Elbashir tries to run the following code, but it contains an error so the program stops. On which line of code does the program stop? Why?

Line Number	Code
1	REPEAT 2 times:
2	move 4 squares right
3	pick up bone
4	move 2 squares left
5	put down bone
6	end REPEAT
7	move 1 square left
8	pick up bone
9	move 3 squares right
10	put down bone

- (b) Rewrite the code so that the program runs properly and once it's finished, the bones are in the four rightmost squares. As an extra challenge, see if you can do this using only 10 lines of code.

Solution

- (a) To find the line with the error we will trace through the code. We number the squares from 1 to 10, starting on the left, and mark Scruffy's position with a paw print. The bones are initially in squares 2, 5, 6, and 7, and Scruffy is in square 1.





The first line tells Scruffy to move 4 squares right to square 5, which has a bone. He picks up the bone and then moves 2 squares left to square 3, which is empty. He puts down the bone. Thus, after the first time through the repeat block, the bones are in squares 2, 3, 6, and 7, and Scruffy is in square 3.



We then go through the repeat block a second time. From square 3, Scruffy moves 4 squares right to square 7, which has a bone. He picks up the bone and moves 2 squares left to square 5, which is empty. He puts down the bone. Thus, after the second time through the repeat block, the bones are in squares 2, 3, 5, and 6, and Scruffy is in square 5.



From square 5, line 7 of the code tells Scruffy to move 1 square left to square 4, which is empty. The next line of code says to pick up a bone, but since square 4 is empty, this results in an error and causes the program to stop. So the program stops on line 8.

- (b) There are many ways to rewrite the code so that the bones are in the four rightmost squares after the program runs. An example is shown to the right.

The bones in squares 2, 5, and 6 need to be moved to squares 8, 9, and 10, in any order. To move each bone it takes 4 lines of code because Scruffy needs to move to a square with a bone, pick up the bone, move to an empty square, and put down the bone. Thus, if each line of code is executed once, then it would take 12 lines of code in total.

```

move 1 square right
pick up bone
move 8 squares right
put down bone
move 5 squares left
pick up bone
move 4 squares right
put down bone
move 3 squares left
pick up bone
move 2 squares right
put down bone
    
```

If we want to move the 3 bones using only 10 lines of code, we will need to make use of a REPEAT block. An example is shown to the right.

```

move 1 square right
pick up bone
move 6 squares right
put down bone
REPEAT 2 times:
    move 3 squares left
    pick up bone
    move 4 squares right
    put down bone
end REPEAT
    
```