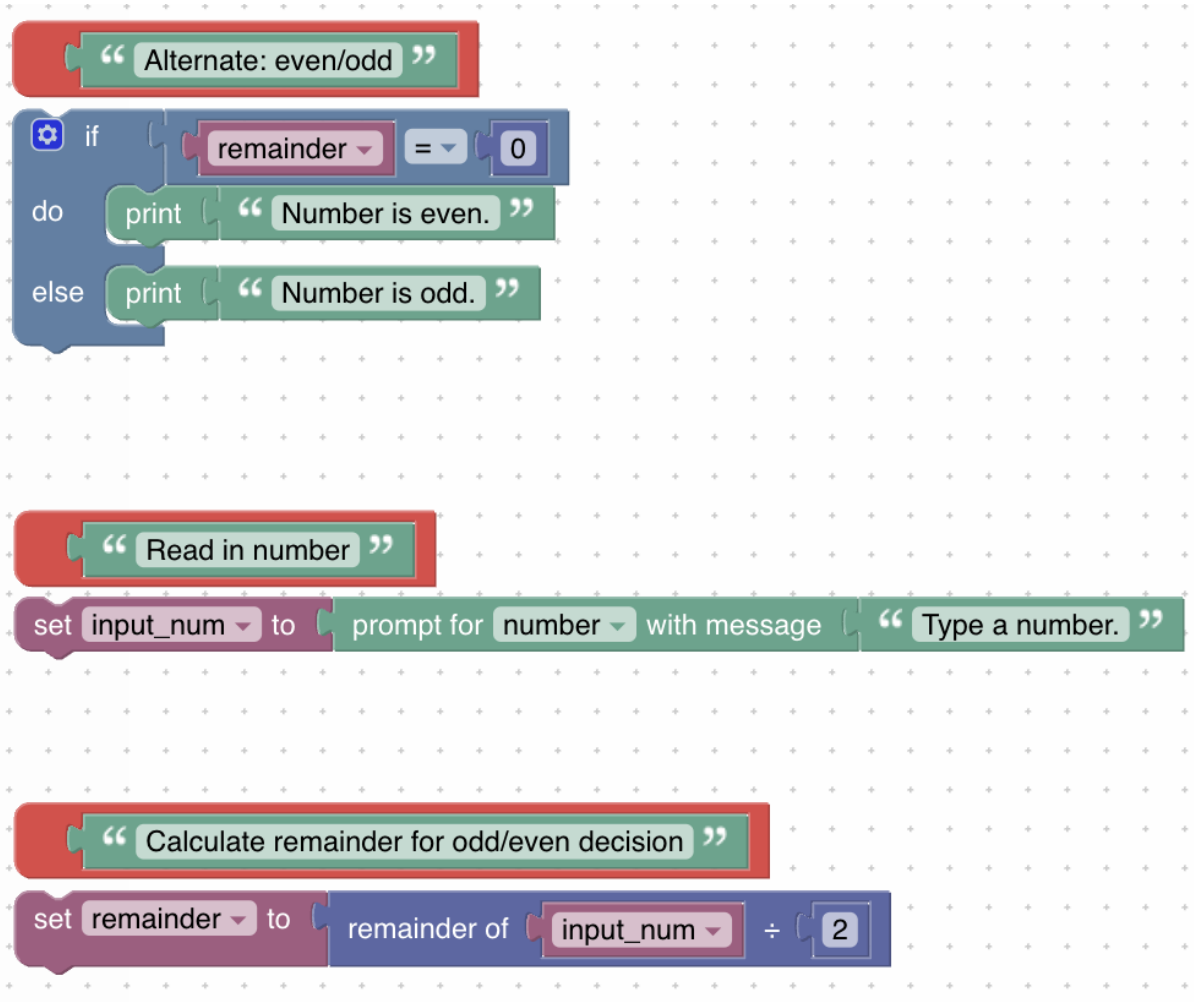
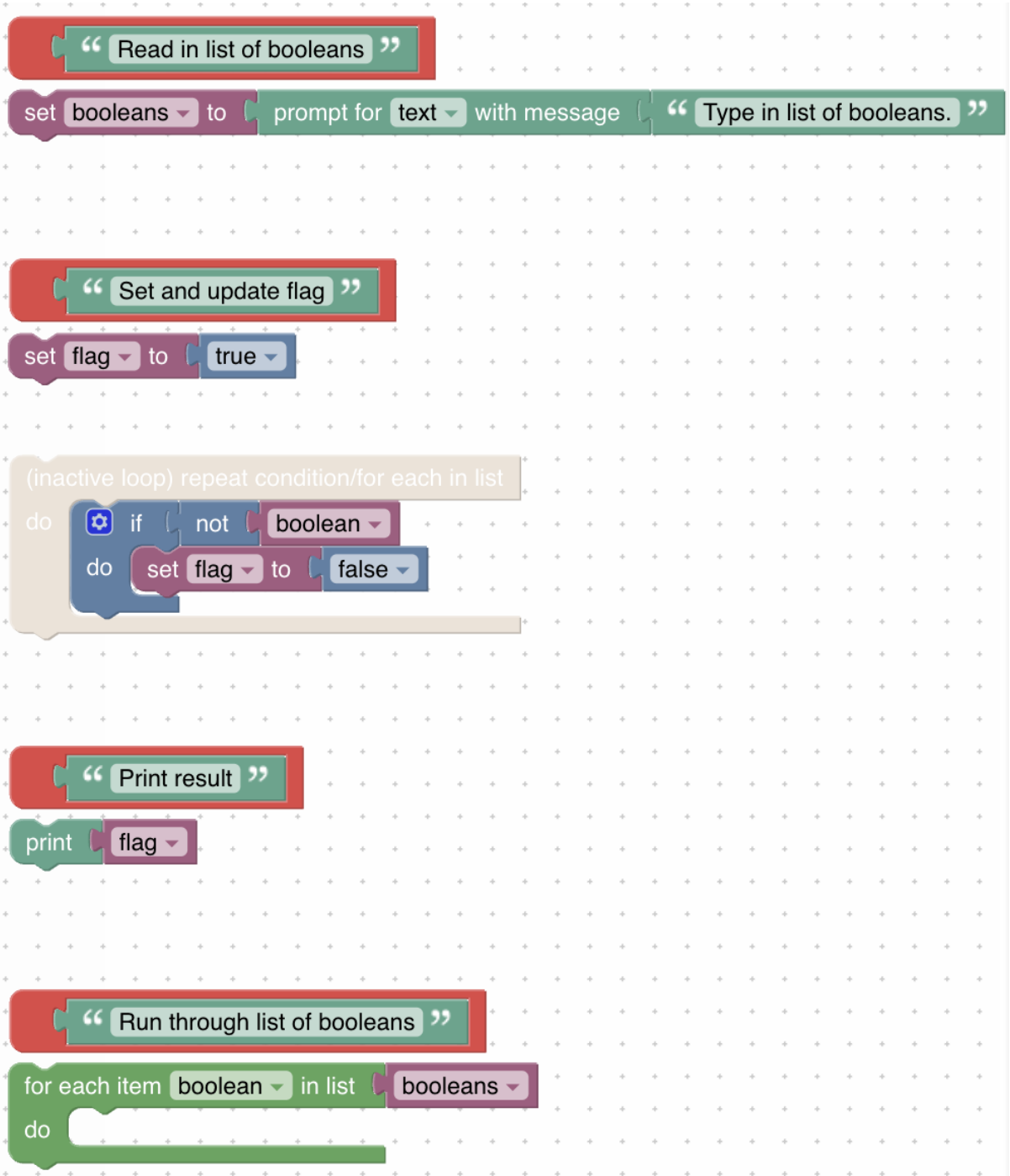
Even or odd (easy)

Report whether an Input number is even or odd.



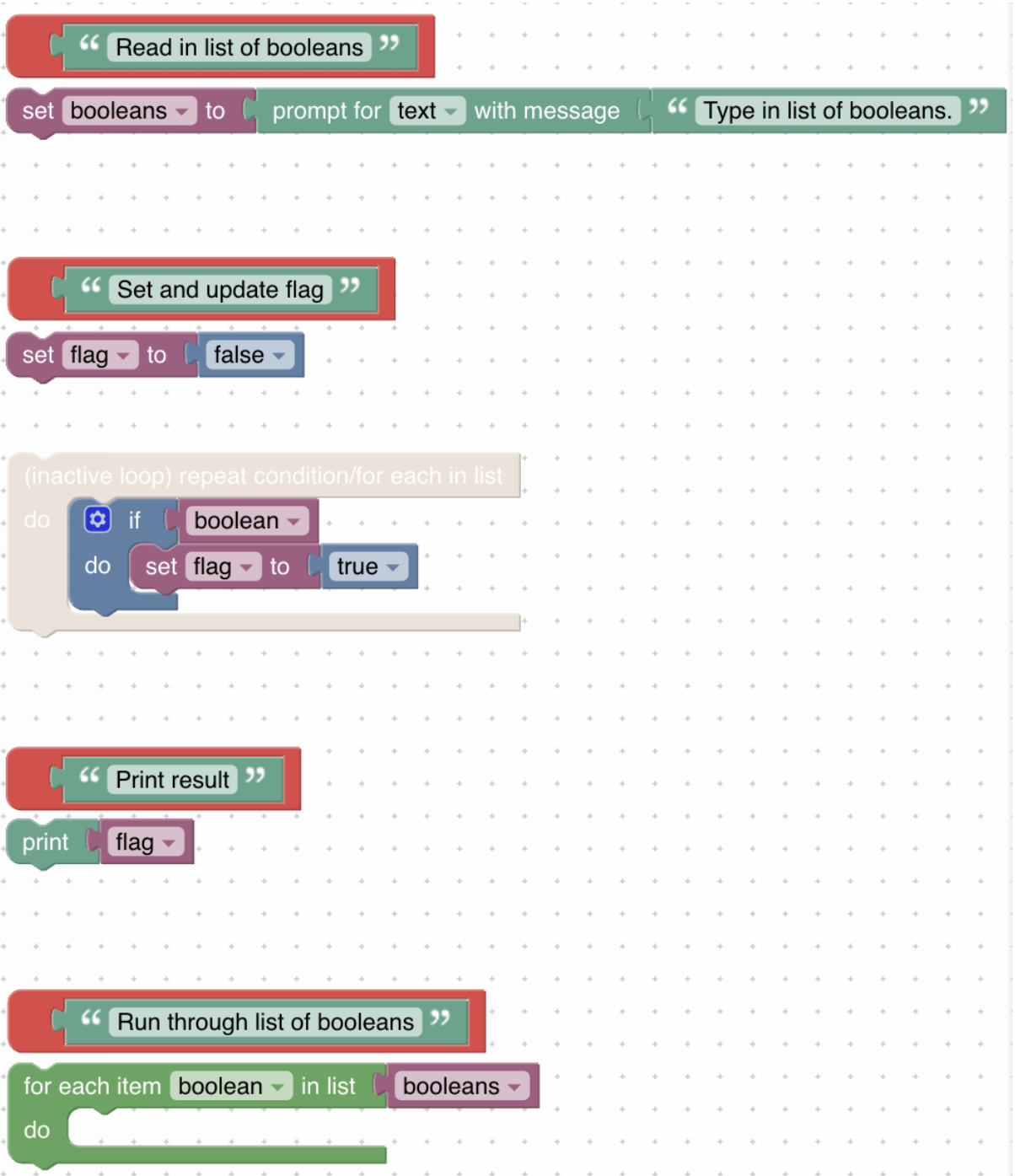
Logical AND (easy)

Write a program that logically evaluates the AND for a list of booleans. If and only if ALL values are True, the program prints True, otherwise False.



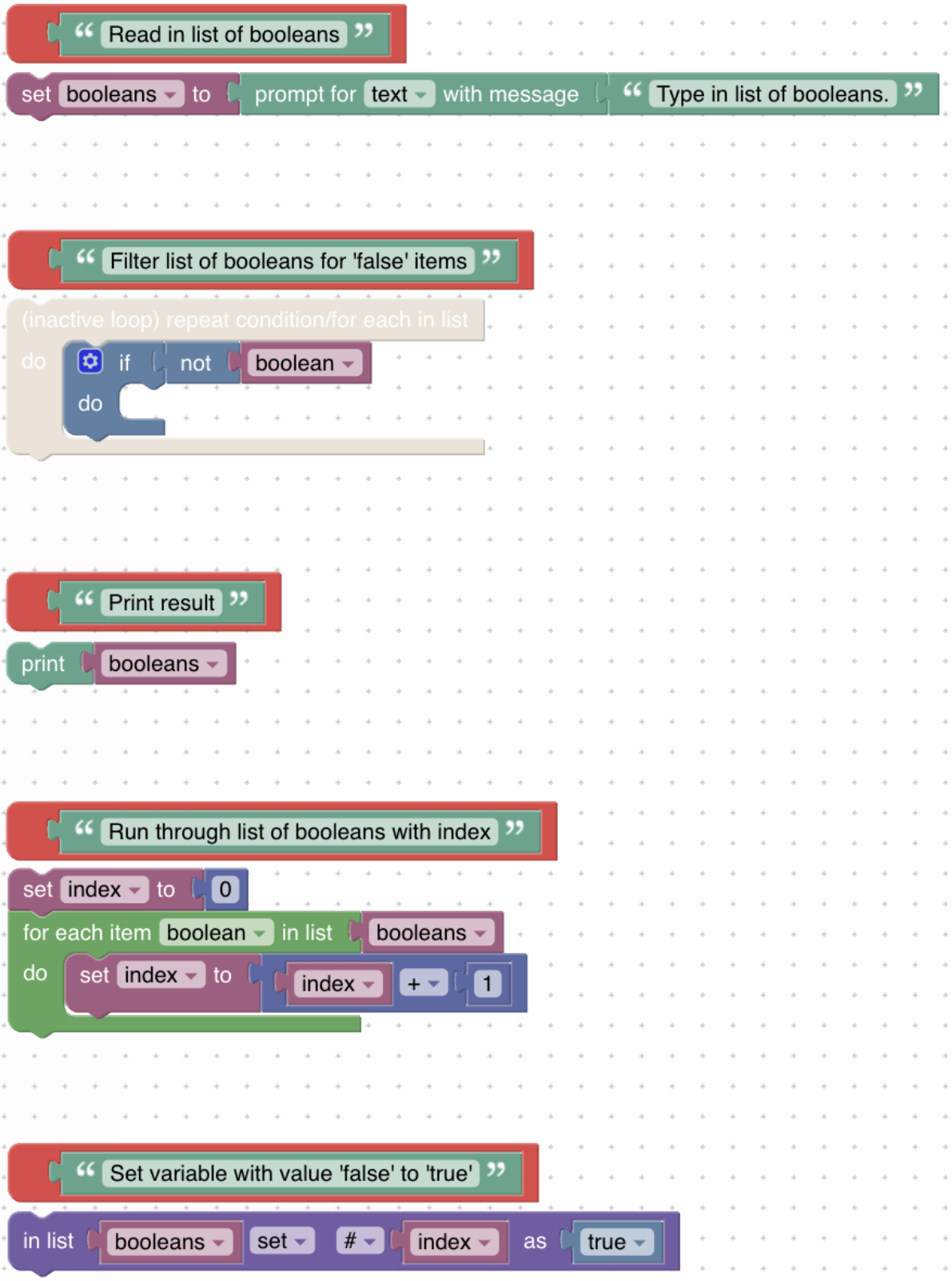
Logical OR (easy)

Write a program that logically evaluates the OR for a list of booleans. If at least one value is True, the program prints True, otherwise False.



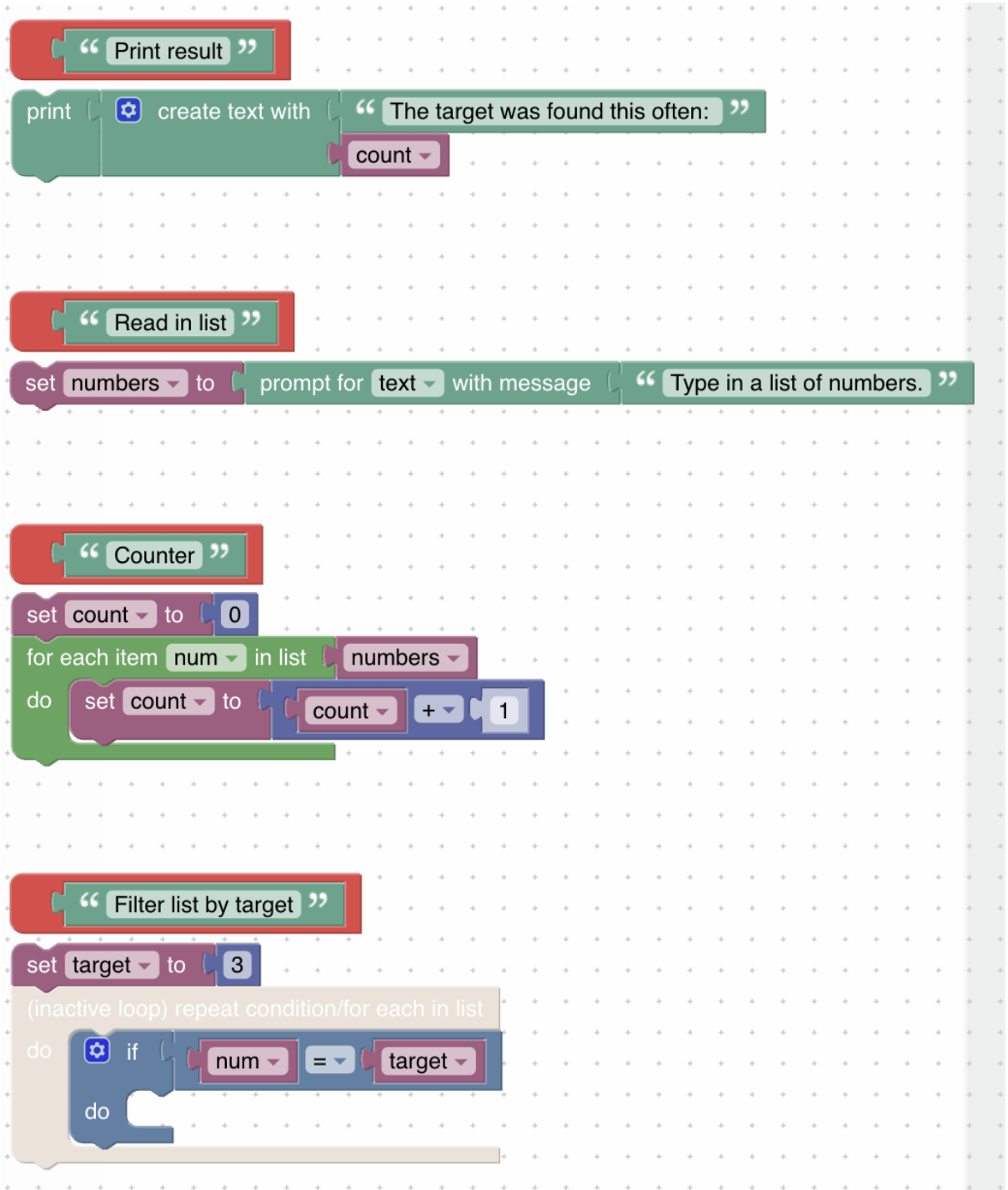
Replace falses (easy)

Write a program that reads in a list of booleans and replaces all occurrences of the value False by True For example when reading the list [True,True,False] you should in the end print the list [True,True,True].


“ Read in list of booleans ”
set booleans to prompt for text with message “ Type in list of booleans. ”
“ Filter list of booleans for 'false' items ”
(inactive loop) repeat condition/for each in list
do if not boolean
do
“ Print result ”
print booleans
“ Run through list of booleans with index ”
set index to 0
for each item boolean in list booleans
do set index to index + 1
“ Set variable with value 'false' to 'true' ”
in list booleans set # index as true

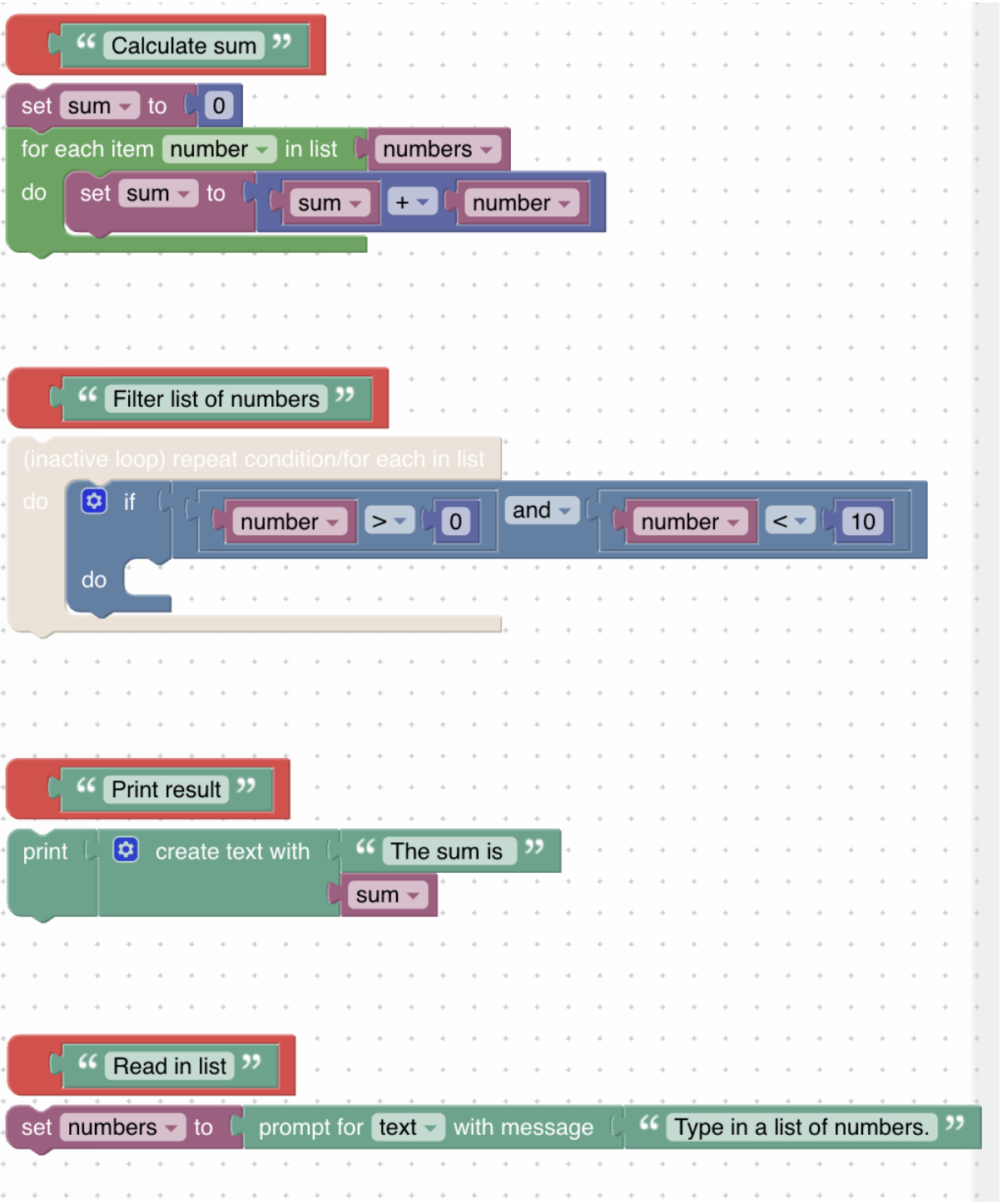
Count target values (easy)

One of the common things to do with a list is to count the number of times a target value appears in a list. Write a program that counts the occurrences of the target value 3 in a list you read in.
For example the program should print out - The target was found this often: 0 - for the list [1, 2, 5] or - The target was found this often: 2 - for the list [3, 4, 3].



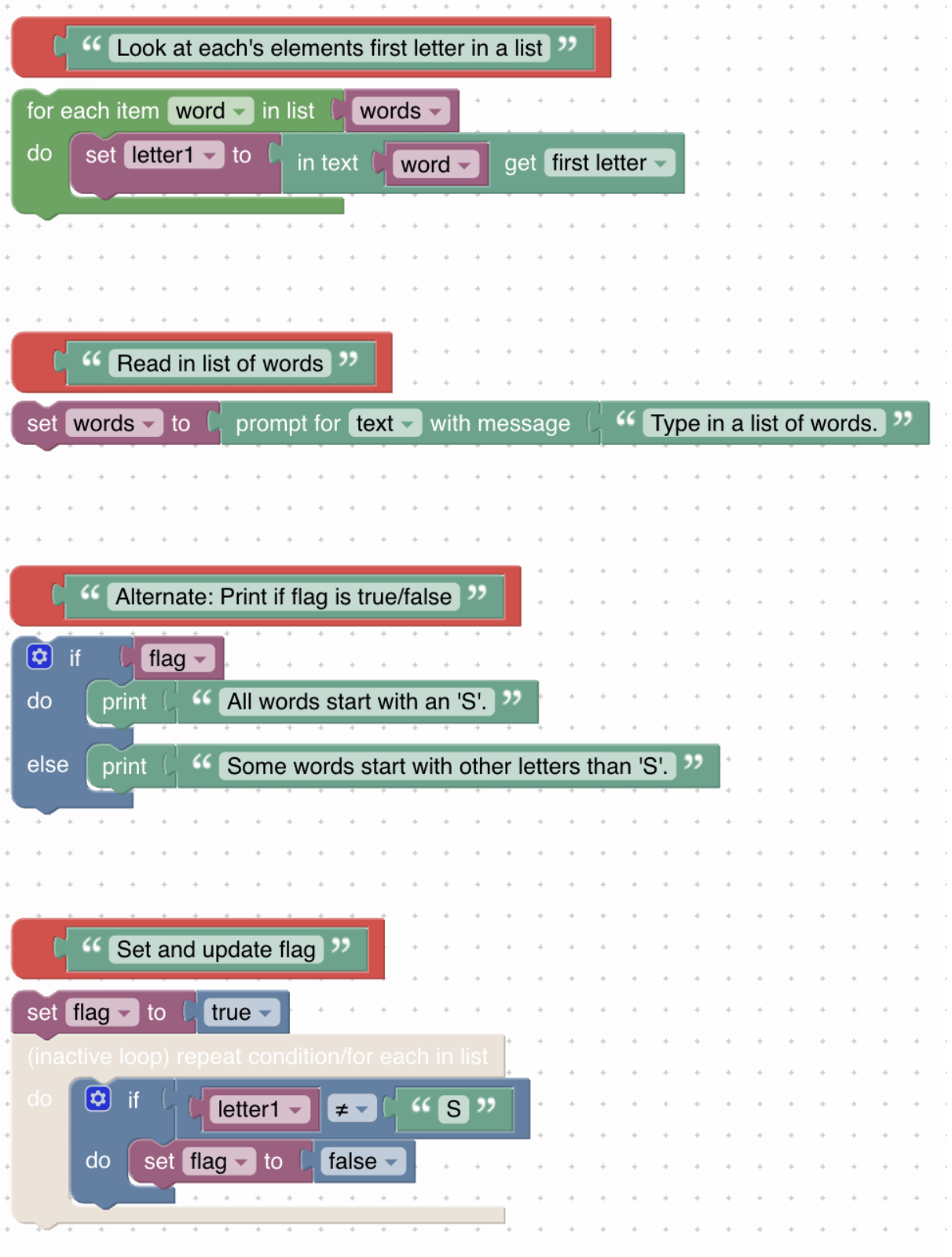
Filtered sum (easy)

Construct a program that reads in a list of numbers and prints the sum of all values in the list that are single digit positive numbers. Loop through all values in the list and add each one that matches the condition to the sum. Print the sum.



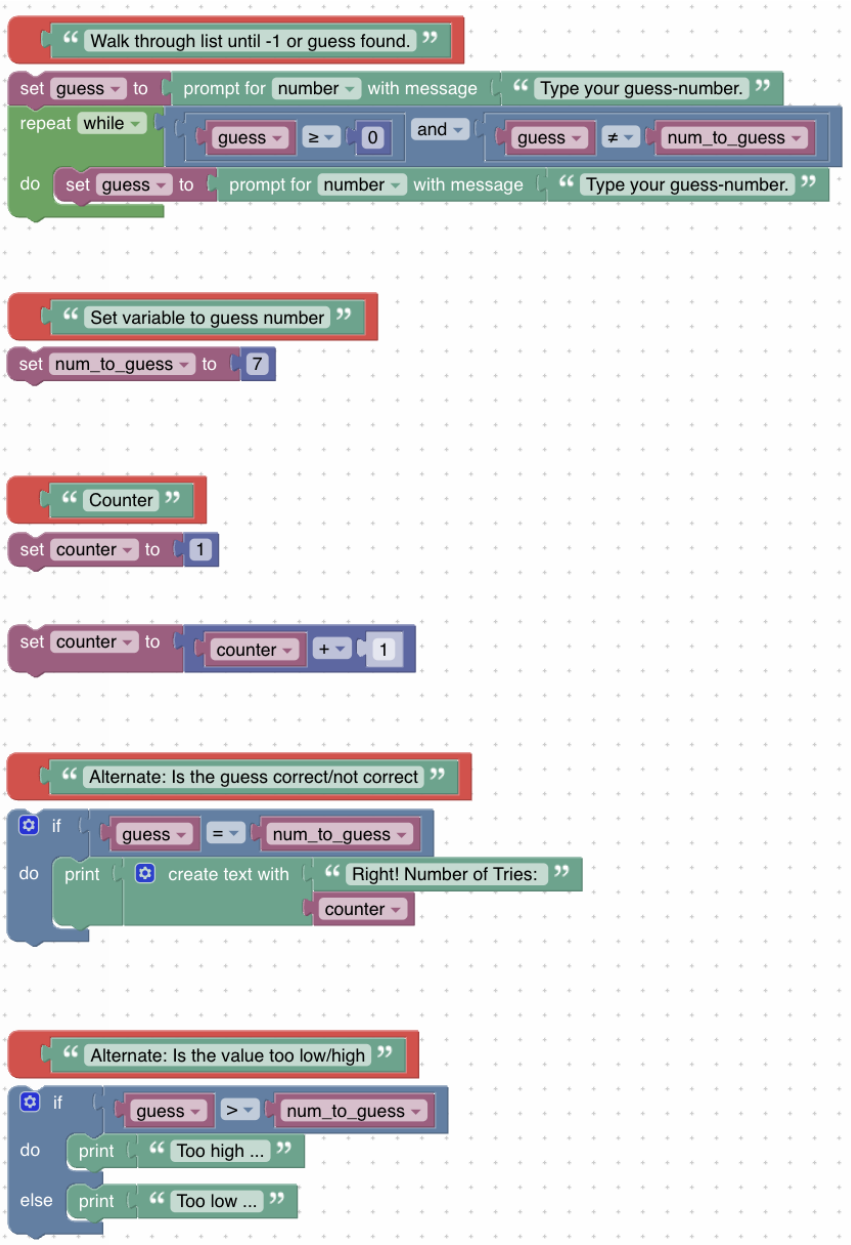
Word start checker (easy)

Construct a program that reads in a list of words and prints out if all words start with the letter S or not. You do not have to worry about upper or lower case – only look for uppercase S.



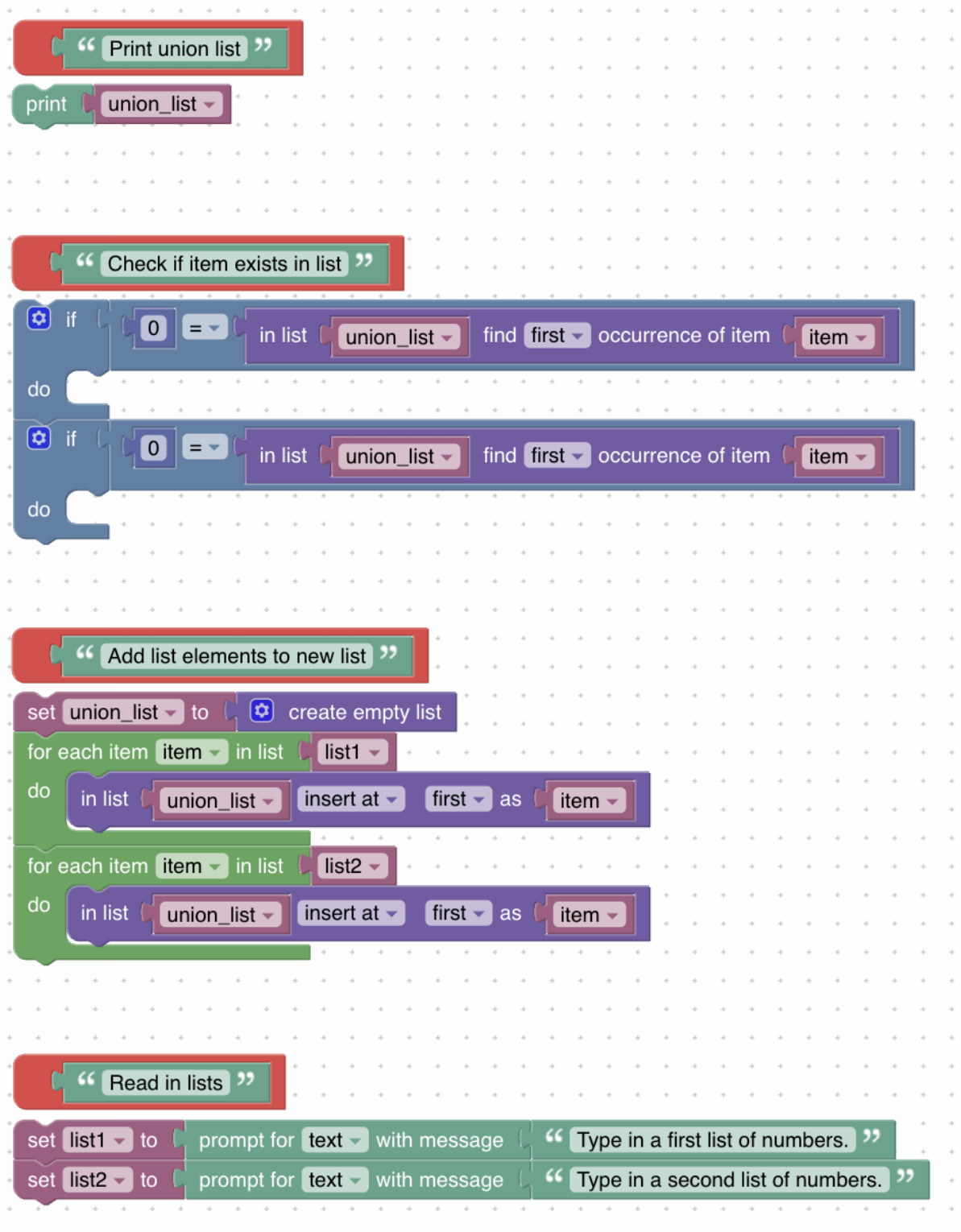
Guessing game (medium)

First we set a guess number (in this case it is set to 7). Now write a program that reads in numbers and ends immediately when a negative number is read or the guess was correct. Then check if each value matches the guess number. The program should also give feedback if the Input is too low or too high and in the end how many tries it took.



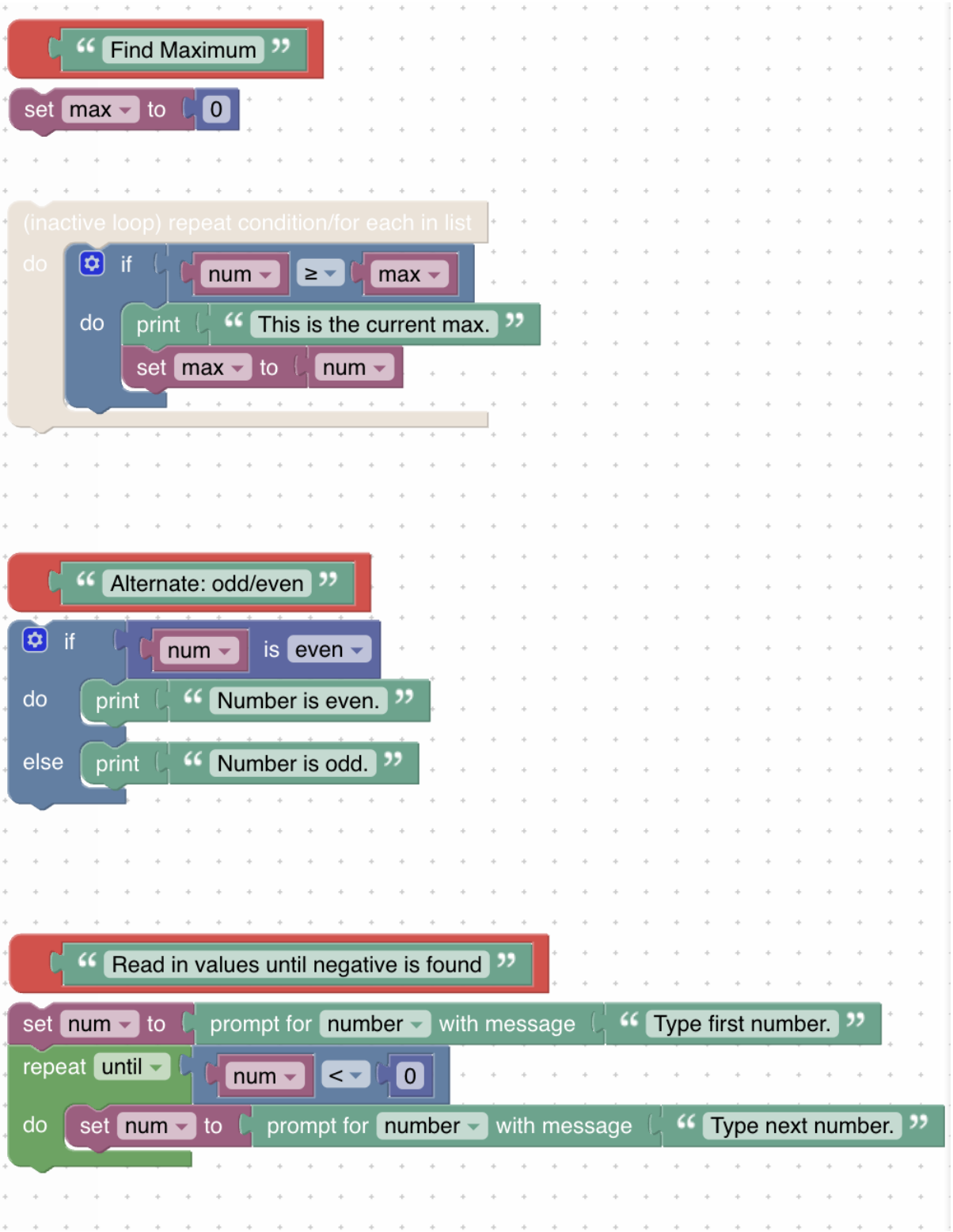
Union of lists (medium)

Write a program that reads in two lists and prints the union of both lists.
For example when given the two lists [5,3,7,-1] and [3,7,-5,0] the program prints '[5,3,7,-1,-5,0]'.
Note: The function 'in list [union_list] find first occurrence of item [item]' returns 0 if the item is not found.



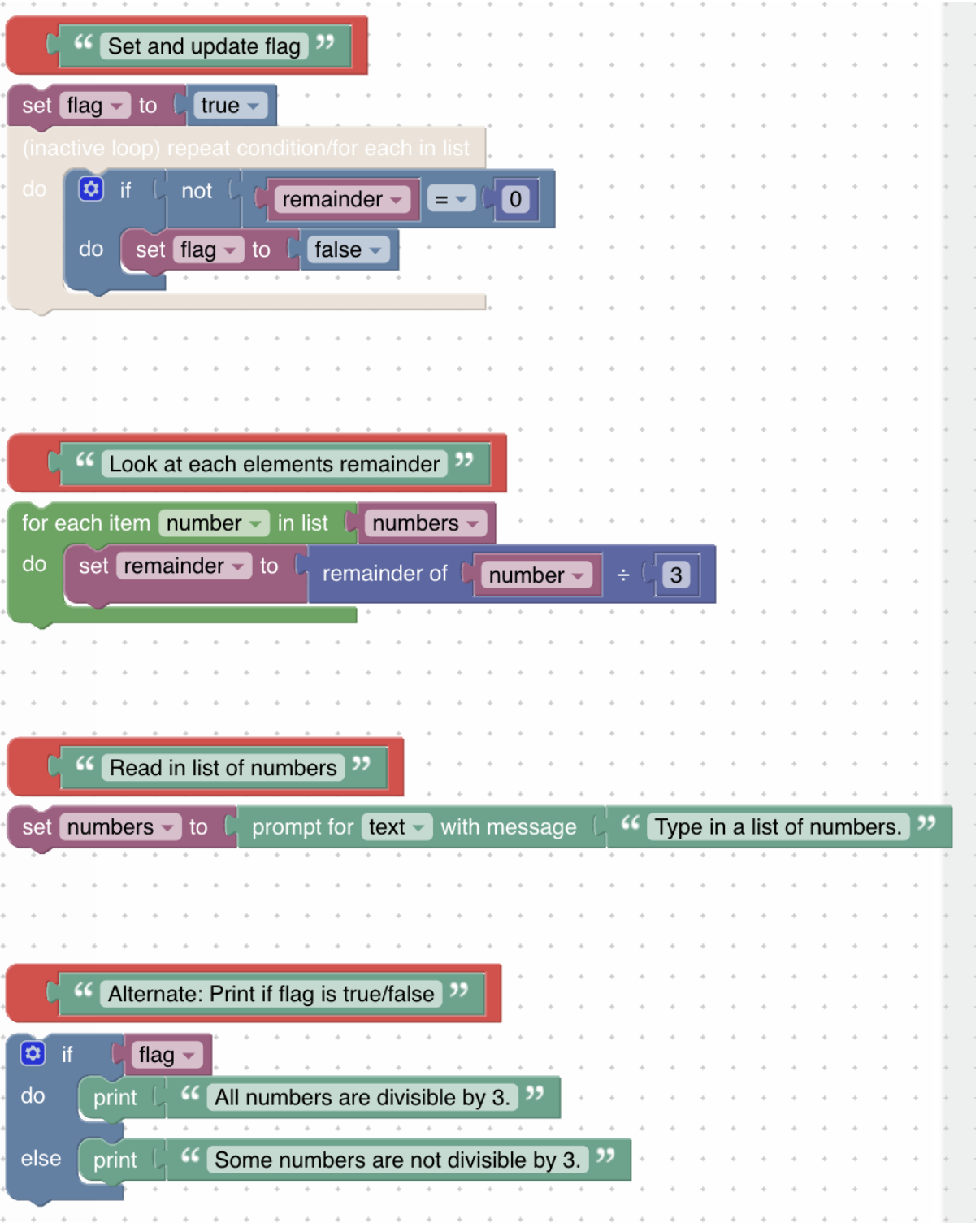
Find even or odd and maximum in list (medium)

Write a program that reads in numbers and ends immediately when a negative number is read. For all the numbers in the list it should output first if the number is even or odd and afterwards whether it is the current maximum.



Word start checker B/ Divisible by 3 (easy)

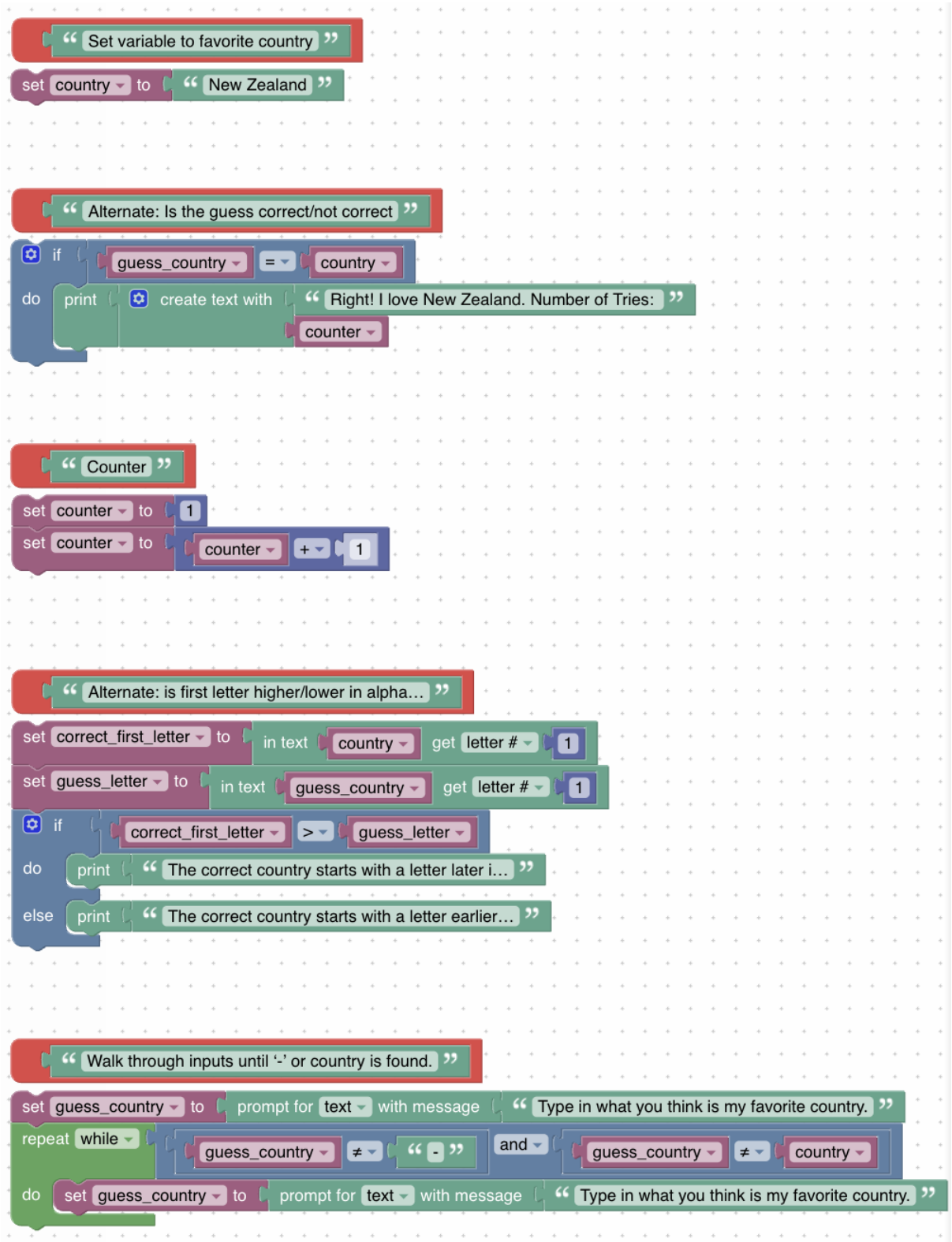
Construct a program that reads in a list of numbers and prints out if all numbers are divisible by 3 (remainder 0) or not.



Guessing game B/ Guessing Country (easy)

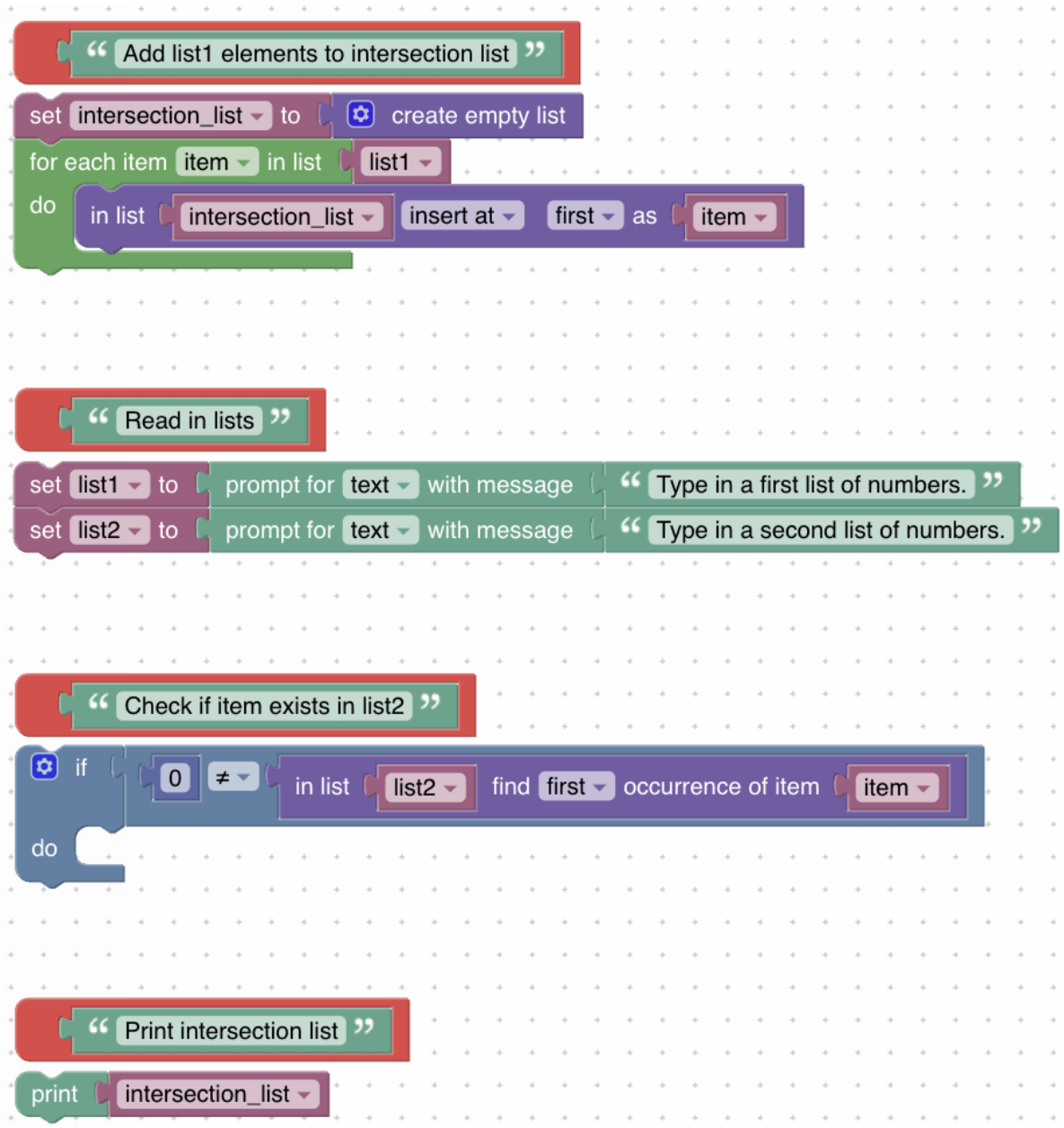
This program should read suggestions for what you think is my favourite country. First we set my favourite country (Psst ... it is New Zealand). Now write a program that reads in words and ends immediately when a '-' is read or the guess was correct. Then check if the guess value matches my favourite country. You do not have to worry about upper or lower case, every word will start with an

uppercase letter and have otherwise lowercase ones. The program should also give feedback if the guessed country has its first letter later or earlier in alphabet compared to the correct country and in the end how many tries it took.



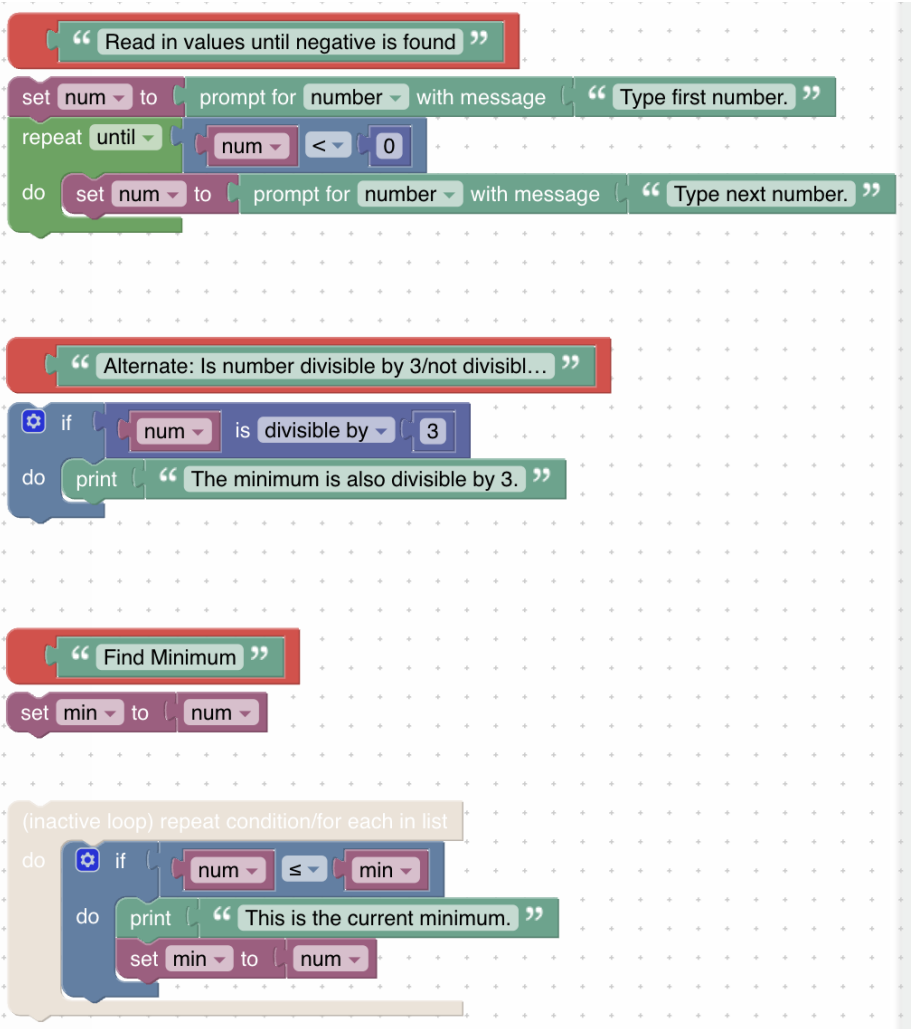
Union of lists B/ Intersection of lists (medium)

Write a program that reads in two lists and prints the intersection of both lists.
For example when given the two lists [5,3,7,-1] and [3,7,-5,0] the program prints '[7, 3]'.
Note: The function 'in list [list2] find first occurrence of item [item]' returns 0 if the item is not found.



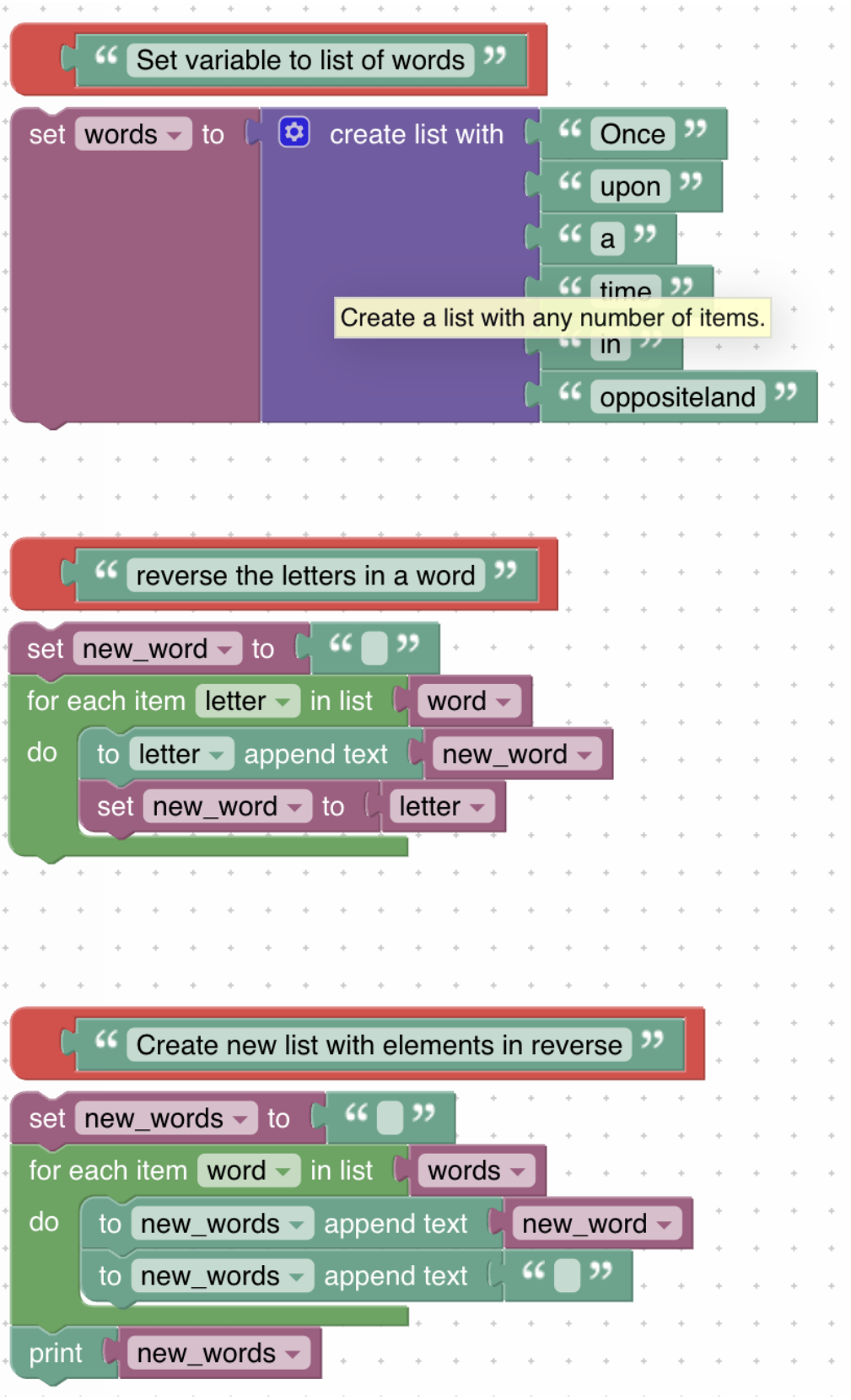
Find even or odd and maximum in list B / Find min and divisible by 3 (medium)

Write a program that reads in numbers and ends immediately when a negative number is read. For all the numbers in the list it should output 'This is the current minimum.' when it is the current minimum, and when it is, print also if it is divisible by 3.



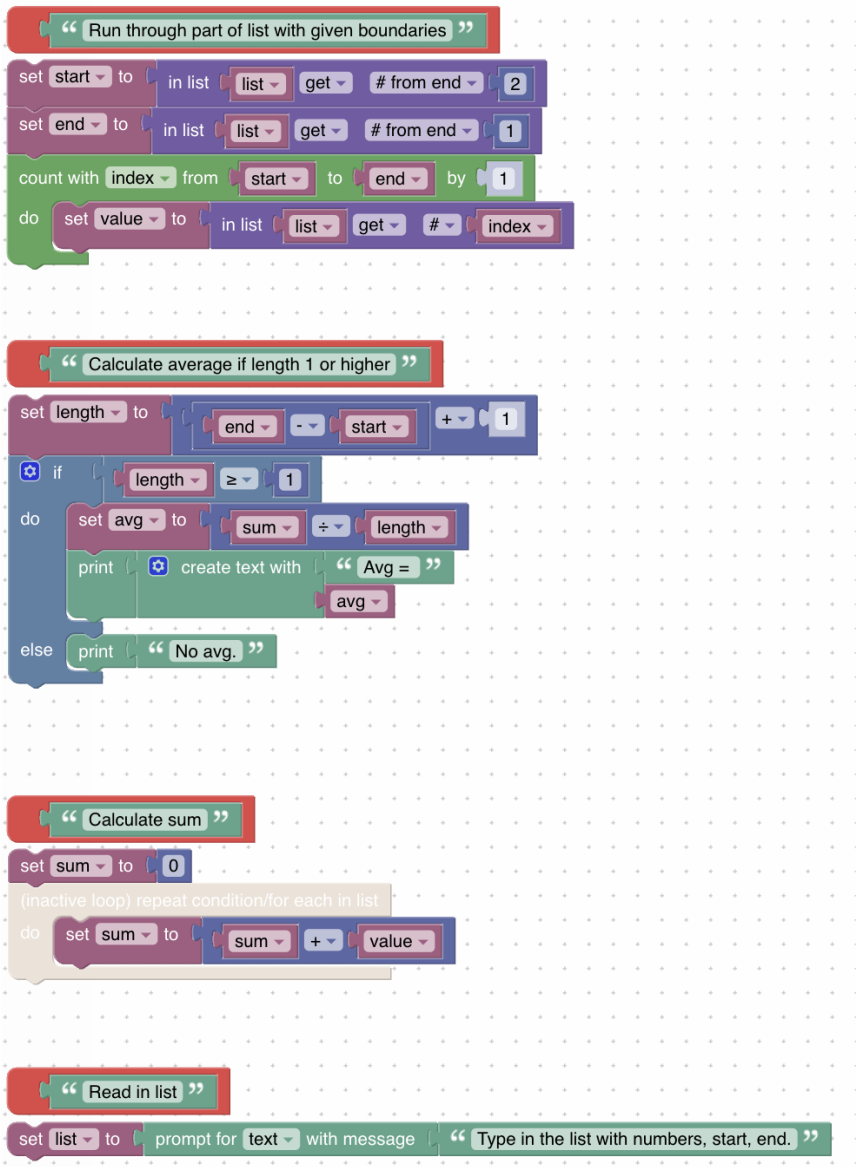
Reverse word and wordlist (medium)

Given a list of words which make up a sentence, generate and print out a string which consists of all the words in reverse order, with each word itself reversed too. Place a space after every word. For example, if the list of words contains “Hello” and “World” in that order, the string created would be “dlroW olleH “.


" Set variable to list of words "
set words to create list with
" Once "
" upon "
" a "
" time "
Create a list with any number of items.
" in "
" oppositeland "
" reverse the letters in a word "
set new_word to " "
for each item letter in list word
do to letter append text new_word
set new_word to letter
" Create new list with elements in reverse "
set new_words to " "
for each item word in list words
do to new_words append text new_word
to new_words append text " "
print new_words

Average value in range. (medium)

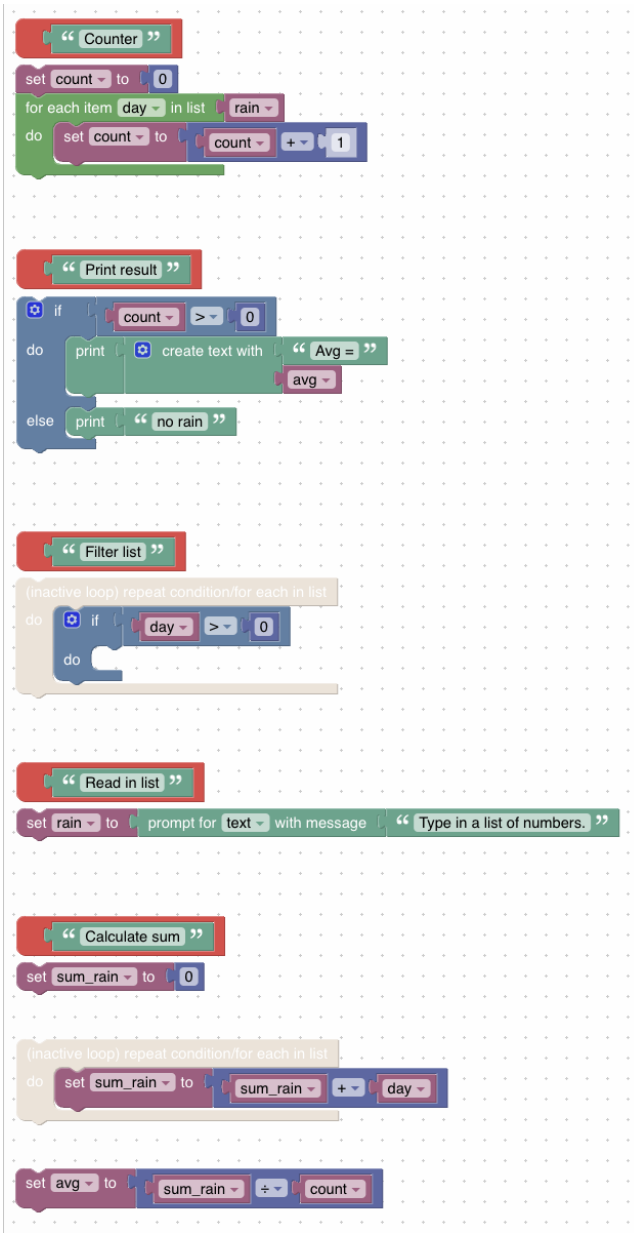
Construct a program that reads a list of numbers. This list consists of a variable number of values followed by an additional number for the start index and a second additional number for the end index of a range of values within the list. The program should print the average of the values between the start and end indices (inclusive). Check if the difference between the end index and the start index is 1 or higher. If so, print the average of the values from start to end and otherwise print 'No avg.' (to prevent a divide by 0).
For example the list [5,3,7,1,7,15,30,2,5] with start 2 and end 5 should print 7.5. NOTE: the indices start from 0 and the end value is included (also not like Python).



Rainfall problem (advanced)

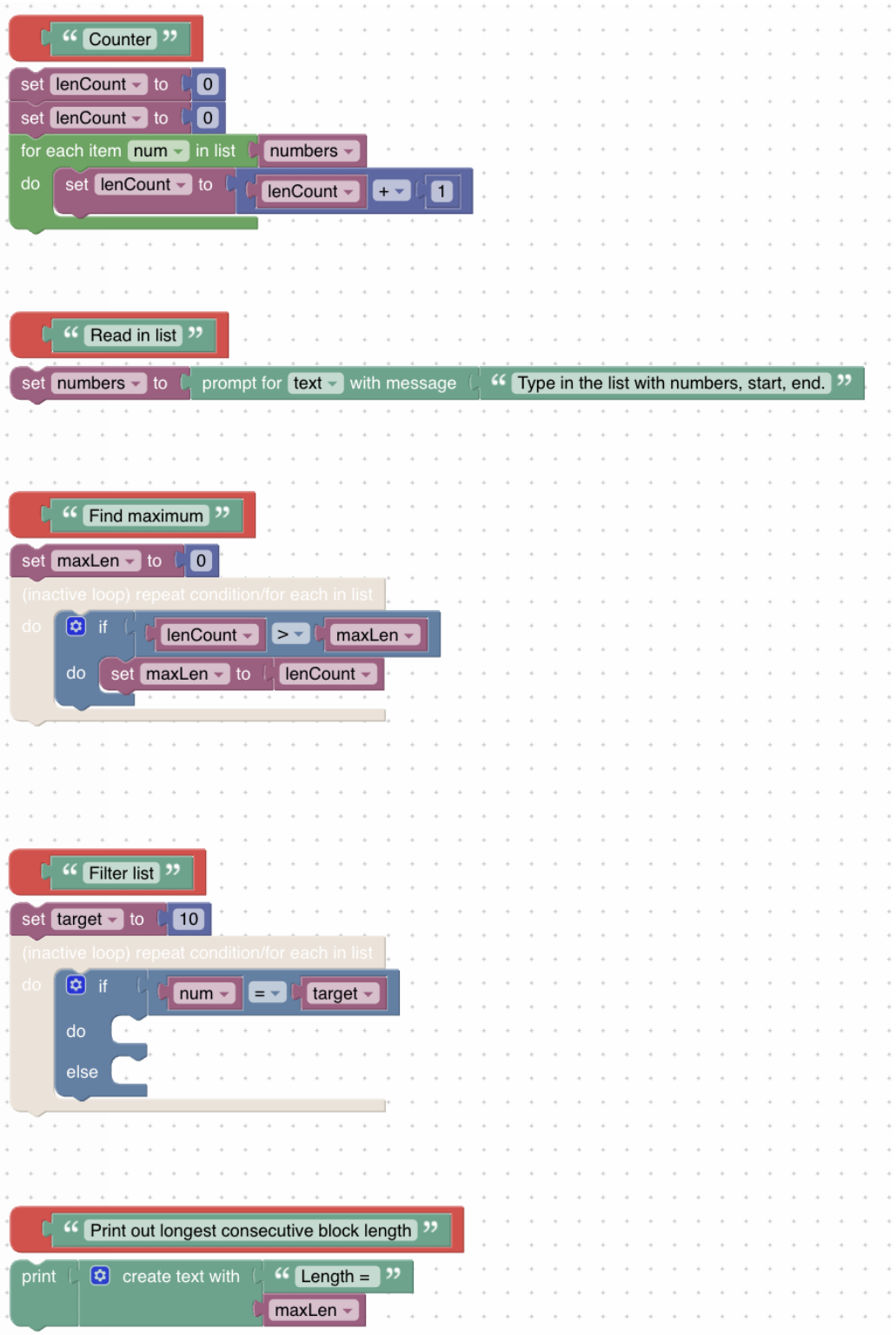
Let’s imagine that you have a list that contains amounts of rainfall for each day, collected by a meteorologist. Her rain gathering equipment occasionally makes a mistake and reports a negative

amount for that day. We have to ignore those. We need to write a program to (a) calculate the total rainfall by adding up all the positive integers (and only the positive integers), (b) count the number of positive integers (we will count with “1.0” so that our average can have a decimal point), and (c) print out the average rainfall at the end. Only print the average if there was some rainfall, otherwise print “No rain”.



Find longest (advanced)

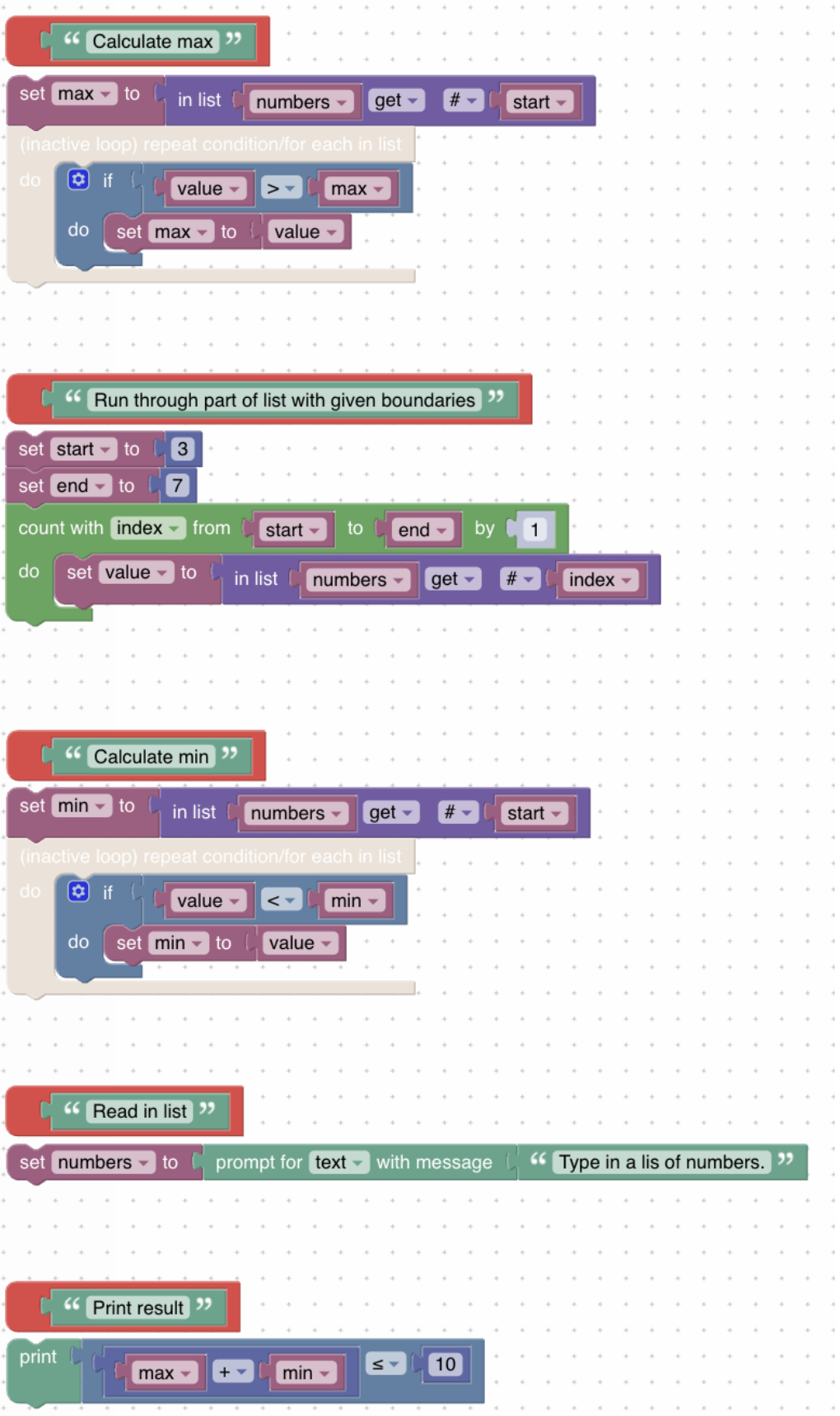
Write the find-longest program, which aims to find the longest consecutive block of the value target (in this case 10) occurring in the read-in list numbers.
For example, if the list numbers contain the values [7, 10, 10, 15, 15, 15, 15, 10, 10, 10, 15, 10] the program should print 3, the length of the longest consecutive block of 10s.



Is level trail segment (advanced)

A trail segment (for running, walking, cycling, etc) is a list of numbers (known as markers) representing a sequence of heights along the trail. There is always a start and an end marker, and maybe more markers in between.

Write a program to read a list of numbers for a trail segment and print True if the segment is “level”and false otherwise. A trail segment is level if the difference between the maximum and minimum elevation on the segment is less than or equal to 10 metres.


“ Calculate max ”
set max to in list numbers get # start
(inactive loop) repeat condition/for each in list
do if value > max
do set max to value
“ Run through part of list with given boundaries ”
set start to 3
set end to 7
count with index from start to end by 1
do set value to in list numbers get # index
“ Calculate min ”
set min to in list numbers get # start
(inactive loop) repeat condition/for each in list
do if value < min
do set min to value
“ Read in list ”
set numbers to prompt for text with message “ Type in a lis of numbers. ”
“ Print result ”
print max + min ≤ 10