## Even or odd (easy)

Report whether an Input number is even or odd.

```
"Alternate: even/odd "

do print "Number is even. "

else print "Number is odd. "

set input_num to prompt for number with message "Type a number. "

set remainder to remainder of input_num 

to remainder 

to
```

## 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.

```
Read in list of booleans
set booleans
                                                           Type in list of booleans.
                       prompt for text
        Set and update flag
set flag -
                 true
                      boolean
           set flag
                            false
        Print result 22
        flag
        Run through list of booleans 22
for each item boolean -
                                  booleans -
```

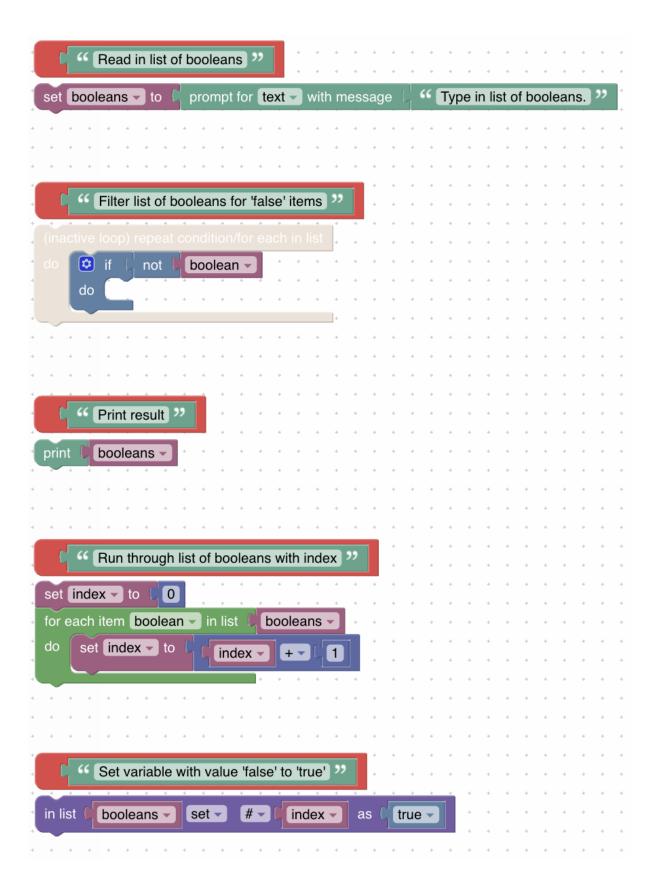
# Logical OR (easy)

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.

```
Read in list of booleans 22
                                                           Type in list of booleans.
set booleans
                       prompt for text
        Set and update flag
set flag -
                 false
               boolean -
           set flag -
                            true
        Print result
        flag
        Run through list of booleans
for each item boolean
                                  booleans
```

# 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].



## 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].

```
Print result 22
                             The target was found this often:
        create text with
                             count -
       Read in list
set numbers
       Counter 33
set count to
for each item num -
                            numbers
                         count -
       Filter list by target
set target -
     num 🔻
                                target -
```

## 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.

```
Calculate sum
set sum v to
for each item number -
                                numbers -
     set sum 		 to
                        sum 🔻
        Filter list of numbers
     number -
                                                       number -
     do
        Print result 22
                              "The sum is
        create text with
        Read in list
                                                       Type in a list of numbers.
set numbers
```

## 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.

```
Look at each's elements first letter in a list
for each item word v in list
                                 words
      set letter1 ▼
                           in text
                                     word -
                                                get first letter
         Read in list of words
set words
                     prompt for text
                                        with message
                                                              Type in a list of words.
        Alternate: Print if flag is true/false
🔯 if
            flag 🕶
                All words start with an 'S'.
                Some words start with other letters than 'S'.
else
         Set and update flag
set flag
                  true
      *
                   letter1 ▼
           set flag - to
```

## 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.

```
Walk through list until -1 or guess found.
set guess - to
                 prompt for number with message
                                                     "Type your guess-number.
repeat while
    set guess
    Set variable to guess number
   num_to_guess - to (
    Counter 2
 et counter - to
                    counter
  Alternate: Is the guess correct/not correct
                          num_to_guess -
                                 "Right! Number of Tries:
                                counter
    Alternate: Is the value too low/high
                            num_to_guess
```

### 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.

```
Print union list 22
        union_list
        Check if item exists in list 22
🔅 if
                         in list
                                                 find first v occurrence of item
                                  union list -
0
                         in list
                                                 find first v occurrence of item
        Add list elements to new list 22
set union_list v to
                        create empty list
for each item item
                               list1 🕶
      in list
                             insert at 🕶
              union_list -
for each item item in list
                               list2
              union_list -
                             insert at 🕶
        Read in lists
set list1 -
                  prompt for text
                                     with message
                                                        Type in a first list of numbers.
                  prompt for text
                                                        Type in a second list of numbers.
```

## 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.

```
Find Maximum
set max
                             max 🔻
    do
                 "This is the current max.
       Alternate: odd/even
🤨 if
            num 🔻
                    is even
do
             Wumber is even.
             Number is odd.
else
       Read in values until negative is found
               prompt for number
repeat until -
                 num 🔻
                              number
```

# 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.

```
Set and update flag
set flag
                 true
                         remainder -
           set flag -
                            false
        Look at each elements remainder
for each item number
                             remainder of
                                             number
        Read in list of numbers
        Alternate: Print if flag is true/false
flag
do
               All numbers are divisible by 3.
                  Some numbers are not divisible by 3.
```

# 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.

```
Set variable to favorite country
                    66 New Zealand 22
    "Alternate: Is the guess correct/not correct "
           guess_country = = =
                                     country -
                                   "Right! I love New Zealand. Number of Tries:
                                   counter
    "Counter"
set counter - to
                    1
set counter - to
                      counter -
    44 Alternate: is first letter higher/lower in alpha... 22
set correct_first_letter -
                                                   get letter # -
                                       country -
set guess_letter - to
                                                     get letter# -
                                 guess_country -
🤨 if
             correct_first_letter -
                                         guess_letter -
               The correct country starts with a letter later i...
               The correct country starts with a letter earlier...
    Walk through inputs until '-' or country is found.
set guess_country v to prompt for text v with message
                                                             "Type in what you think is my favorite country."
repeat while
                                                           and 🕶
                                                " 🔳 "
                      guess_country -
                                                                       guess_country ▼ ≠ ▼
                                                                                                 country -
do set guess_country v to
                                prompt for text -
                                                                   Type in what you think is my favorite country.
```

## 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.

```
44 Add list1 elements to intersection list 32
set intersection_list - to
                              create empty list
for each item item -
                                list1 🕶
               intersection_list -
                                    insert at 🕶
                                                  first -
                                                                item -
         Read in lists 22
                  prompt for text -
                                                         Type in a first list of numbers.
set list1 -
            to
                                      with message
                                                         Type in a second list of numbers.
set list2
                  prompt for text
         Check if item exists in list2 22
                         in list
                                  list2 ▼
                                            find first
                                                                               item 🕶
do
         Print intersection list 22
        intersection_list
```

# 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.

```
set num to prompt for number with message "Type first number."

repeat until num vito prompt for number with message "Type next number."

with message "Type next number."

Alternate: Is number divisible by 3/not divisibl...

if num is divisible by 3

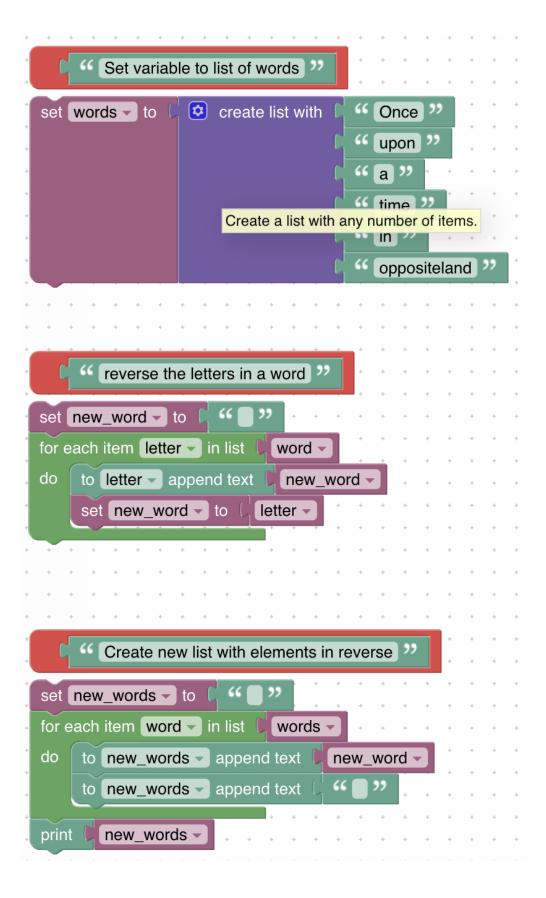
do print "The minimum is also divisible by 3."

(nactive loop) repeat condition/for each in list do print "This is the current minimum."

set min to num vito num vito
```

#### 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".



#### 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).

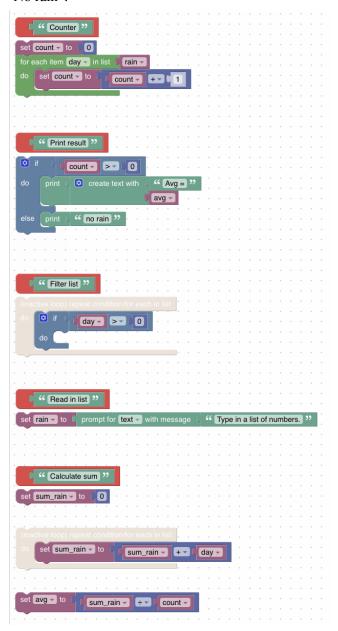
For example the list [5,3,7,1,7,15,30,2,5] with start 2 and end 5 should print 7.5. NOTE: the indices start from 0 and the end value is included (also not like Python).

```
Run through part of list with given boundaries
set start v to
                        list ▼ get ▼ # from end ▼
set end v to
                                      # from end -
count with index - from
                         start -
                      in list ☐ list 		 get 			
    Calculate average if length 1 or higher
set length - to
                                    start -
🤨 if
                                    66 Avg =
                                   avg -
              " No avg. "
    " Calculate sum "
    sum 🕶 to 🏮
     set sum v to
                        sum -
    "Read in list "
                                                 Type in the list with numbers, start, end.
```

#### 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, 10, 15, 10] the program should print 3, the length of the longest consecutive block of 10s.

```
"Counter"
set [lenCount - to
                    0
                    0
set lenCount - to
for each item num -
                            numbers
     set lenCount → to
                            lenCount ▼
    "Read in list
                                                     "Type in the list with numbers, start, end."
    66 Find maximum 33
 et maxLen -
                lenCount →
                                   maxLen
     do set maxLen → to
                              IenCount -
    "Filter list
 et target - to
     🔯 if
                 num 🔻
    "Print out longest consecutive block length"
                             " Length = "
                            maxLen -
```

# 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 "
                                                                                  in list
                                                                                                                          numbers -
                                                                                                                                                                                       get 🕶
                         🔅 if
                                                                                 value 🕶
                                                                                                                                                                max
                                            set max → to
                                                                                                                                     value 🕶
                        "Run through part of list with given boundaries "
set start to 3
set end to
count with index - from
                                                                                                                               start -
                                                                                                                                                                                                    end 🕶
                                                                                                                                                                                                                                                                         1
                        set value va
                                                                                                                 in list
                                                                                                                                                       numbers -
                                                                                                                                                                                                                    get ▼
                                                                                                                                                                                                                                                                                            index -
                        "Calculate min "
 set min → to
                                                                                in list
                                                                                                                        numbers -
                                                                                                                                                                                    get 🕶
                                                                                                                                                                                                                                                             start -
                        🧯 if
                                                                                value 🕶
                                                                                                                                 < ▼
                                                                                                                                                                 min 🕶
                                           set min - to
                                                                                                                                 value 🕶
                       "Read in list "
 set numbers to prompt for text with message 4" Type in a lis of numbers.
                        66 Print result 32
 print
                                                                                                                                                                                                                     10
                                                                                                                                                                                      ≤▼
                                                           max - + -
                                                                                                                                     min 🕶
```