**Problem1:**

Given a list of numbers and a number k, return whether any two numbers from the list add up to k.

For example, given [10, 15, 3, 7] and k of 17, return true since 10 + 7 is 17.

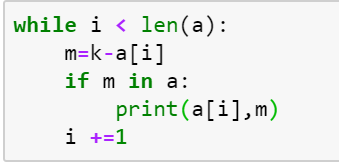
Bonus: Can you do this in one pass?

**Solution:**

a=[10, 15, 3, 7]

k=17

i=0



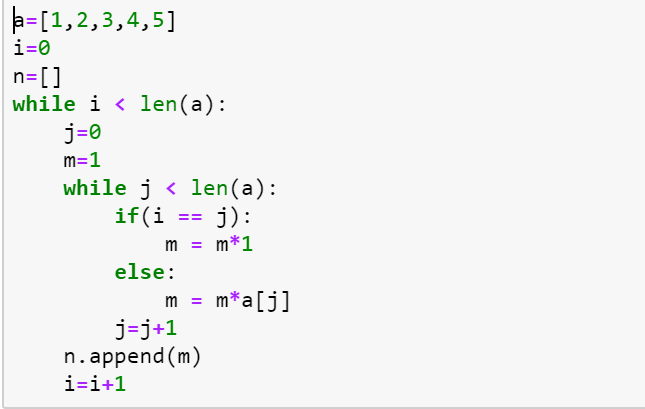
**Problem2:**

Given an array of integers, return a new array such that each element at index i of the new array is the product of all the numbers in the original array except the one at i.

For example, if our input was [1, 2, 3, 4, 5], the expected output would be [120, 60, 40, 30, 24]. If our input was [3, 2, 1], the expected output would be [2, 3, 6].

Follow-up: what if you can't use division?

**Solution:**



**Problem 3**

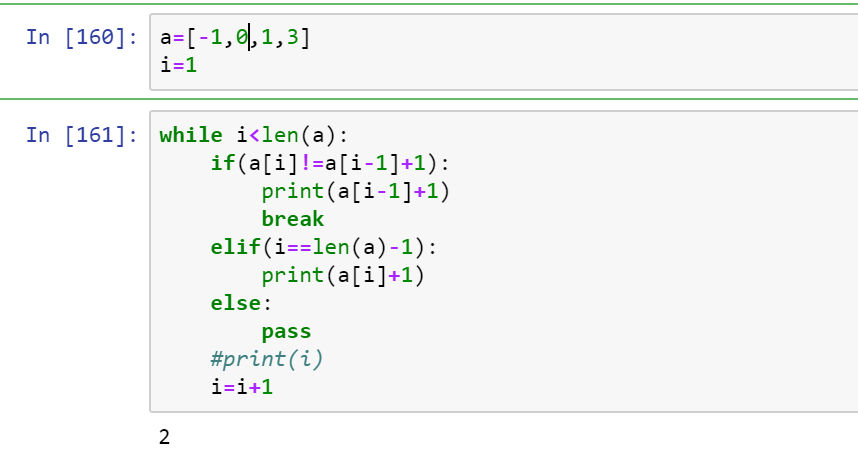
Given an array of integers, find the first missing positive integer in linear time and constant space. In other words, find the lowest positive integer that does not exist in the array. The array can contain duplicates and negative numbers as well.

For example, the input [3, 4, -1, 1] should give 2. The input [1, 2, 0] should give 3.

You can modify the input array in-place.

b=[3,2,1,5,1]

a=list(set(b))



**Problem 5:**

cons(a, b) constructs a pair, and car(pair) and cdr(pair) returns the first and last element of that pair. For example, car(cons(3, 4)) returns 3, and cdr(cons(3, 4)) returns 4.

Given this implementation of cons:

def cons(a, b):

def pair(f):

return f(a, b)

return pair

Implement car and cdr.

