In this version, problem 1b operates on the list ['why', 'me']

problem 1

---------

a) 4

6

9

2 12

b) w

hw

yhw

myhw

emyhw

problem 2

---------

3 [6, 12] [6, 12]

3 [6, 12] [2, 4]

5 [6, 12] [6, 12]

problem 3

---------

a) x | y | output

---------------

0 | 0 | 1

0 | 1 | 0

1 | 0 | 1

1 | 1 | 1

b) \_\_ \_

xy + xy + xy

c) note: you also needed correctly shaped gates

--- ---

|X| |Y|

--- ---

| |

o------- o-------

| | | |

| v | v

| ----- | -----

| |NOT| | |NOT|

| ----- | -----

| | | |

| | | | -------

| |------+------+--->| AND |

| | | | \_\_ |--------

| | o--->| xy | |

| | | ------- | ------

| | | |---->| |

| | | | | ----------

| | | ------- |---->| OR |------>| output |

o-------------+------+--->| AND | | | | ----------

| | | | \_ |-------- |->| |

| | |--->| xy | | ------

| | ------- |

| | |

| | |

| | ------- |

|-------------+---------->| AND | |

| | |----------|

|---------->| xy |

-------

problem 4

---------

def no\_primes(vals):

for x in vals:

if is\_prime(x) == True:

return False

return True

problem 5

---------

00 read r1 # read 1st num

01 read r2 # read 2nd num

02 call r14 05 # call the function

03 write r13 # write the result

04 halt

# the function

05 sub r3 r1 r2 # r3 = r1 - r2

06 jgtz r3 09 # check if r1 > r2

07 copy r13 r2 # if not, r2 is larger

08 jumpn 10 # could also repeat statement 10 here

09 copy r13 r1 # if so, r1 is larger

10 jumpr r14 # return

problem 6

---------

def index\_min\_product(list1, list2):

minpos = 0

minproduct = list1[0] \* list2[0] # NOT 0 or any other constant!

for i in range(len(list1)):

if list1[i] \* list2[i] < minproduct:

minpos = i

minproduct = list1[i] \* list2[i]

return minpos

problem 7

---------

a) 5

9

13

b) a = 2

b = 3

b = 5

a = 9

problem 8

---------

def index\_nth(n, c, s):

count = 0

for i in range(len(s)):

if s[i] == c:

count += 1

if count == n:

return i

return -1