Write a function, sublist, that takes in a list of numbers as the parameter. In the function, use a while loop to return a sublist of the input list. The sublist should contain the same values of the original list up until it reaches the number 5 (it should not contain the number 5).

def sublist(l1):

idx = 0

list2 = []

while idx < len(l1):

if l1[idx] == 5:

break

list2.append(l1[idx])

idx = idx + 1

return list2

list1 = [1,2,3,4,5,6,7]

sublist(list1)

Write a function called check\_nums that takes a list as its parameter, and contains a while loop that only stops once the element of the list is the number 7. What is returned is a list of all of the numbers up until it reaches 7.

def check\_nums(l1):

idx = 0

list2 = []

while idx < len(l1):

if l1[idx] == 7:

break

list2.append(l1[idx])

idx = idx + 1

return list2

list1 = [1,2,3,4,5,6,7,8,9]

check\_nums(list1)

Write a function, sublist, that takes in a list of strings as the parameter. In the function, use a while loop to return a sublist of the input list. The sublist should contain the same values of the original list up until it reaches the string “STOP” (it should not contain the string “STOP”).

def sublist(l1):

idx = 0

list2 = []

while idx < len(l1):

if l1[idx] == "STOP":

break

list2.append(l1[idx])

idx = idx + 1

return list2

list1 = ["Lima", "Loves", "Lemon", "STOP", "Here"]

sublist(list1)

Write a function called stop\_at\_z that iterates through a list of strings. Using a while loop, append each string to a new list until the string that appears is “z”. The function should return the new list.

def stop\_at\_z(l1):

idx = 0

list2 = []

while idx < len(l1):

if l1[idx] == "z":

break

list2.append(l1[idx])

idx = idx + 1

return list2

list1 = ["Lima", "Loves", "Lemon", "z", "Here"]

stop\_at\_z(list1)

Below is a for loop that works. Underneath the for loop, rewrite the problem so that it does the same thing, but using a while loop instead of a for loop. Assign the accumulated total in the while loop code to the variable sum2. Once complete, sum2 should equal sum1.

sum1 = 0

lst = [65, 78, 21, 33]

for x in lst:

sum1 = sum1 + x

print(sum1)

sum2 = 0

idx = 0

while idx < len(lst):

sum2 = sum2 + lst[idx]

idx = idx + 1

print(sum2)

**Challenge:** Write a function called beginning that takes a list as input and contains a while loop that only stops once the element of the list is the string ‘bye’. What is returned is a list that contains up to the first 10 strings, regardless of where the loop stops. (i.e., if it stops on the 32nd element, the first 10 are returned. If “bye” is the 5th element, the first 4 are returned.) If you want to make this even more of a challenge, do this without slicing.

def beginning(l1):

idx = 0

list2 = []

while idx < 10:

if l1[idx] == "bye":

break

list2.append(l1[idx])

idx = idx + 1

return list2

list1 = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'bye', 'k']

beginning(list1)