Nondecreasing

Write a program that asks the user to enter a sequence of "non-decreasing" numbers one at a time. Numbers are non-decreasing if each number is greater than or equal to the last.

When the user enters a number which is smaller than their previously entered value, the program is over. Tell the user how long their sequence was.

Concepts:

This problem mixes variables with control flow. In class we went over the [guess number program](https://edstem.org/us/courses/10000/lessons/13552/slides/67359) which was our introduction to variables with control flow.

This problem required you to use variables in new ways, on the fly. If you were able to solve it, that is certainly a good sign that you have mastered the core concepts. If you knew all of the concepts (loops, variables, input and print) and your challenge was putting it together I have good news, with practice that will become natural.

Details: here were some of the details to look out for:

1. Did you handle the case where the input was equal to the last number? Check if you have an equals sign or not in your condition. Should you?
2. Did you convert your input to a float? This is a real nitpick. But lots of people forgot.
3. Did you cast your count variable to a string before printing it?

Example Solutions



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

"""

This problem is complex because you need to keep track of several variables.

There are so many ways to solve this problem! They generally fall into two main categories.

"""

# Those who used a while loop which goes while curr >= last

def main():

print("Enter a sequence of non-decreasing numbers. ")

count = 0 # essential to keep track of length

last\_num = float(input('Enter num: '))

curr\_num = last\_num # can have another enter num here

while curr\_num >= last\_num: # there are many ways to write this!

count += 1 # this line can go anywhere in body

last\_num = curr\_num # update the last num first!

curr\_num = float(input('Enter num: '))

print("Thanks for playing!") # watch for string

print("Sequence length: " + str(count))

# Those who used a while True loop where they only needed to have a single last\_value

# declared before the loop. This required use of the break keyword.

def main():

print("Enter a sequence of non-decreasing numbers. ")

count = 1 # length is at least 1

last\_num = float(input('Enter num: '))

while True:

curr\_num = float(input('Enter num: '))

if curr\_num < last\_num:

# some people optionally put the thanks for playing here

break # this makes python exit the loop immediately

count += 1

last\_num = curr\_num

print("Thanks for playing!")

print("Sequence length: ", count)

AI Hasn't Given Feedback for This Problem

The AI tried to give feedback on Question 4, but unfortunately, it wasn't able to meet its own bar for quality. We hope the AI will be able to make some progress and reach that bar in the near future, but there are no guarantees because this is very new research.

We will update this page if AI is ready, but we're not sure when that might be. New things can take time. For today, you can see what feedback is like without an AI helper.