INTUITOR PYTHON BACKEND DEVELOPER PROGRAMMING CHALLENGE

Instructions

1. In your solution, focus on correctness and efficiency.
2. You are to use Python3 as your programming language.
3. Write clean code.

1. You would like to set a password for a bank account. However, there are three restrictions on the format of the password:

   it has to contain only alphanumeric characters (a−z, A−Z, 0−9);
   there should be an even number of letters;
   there should be an odd number of digits.
   You are given a string S consisting of N characters. String S can be divided into words by splitting it at, and removing, the spaces. The goal is to choose the longest word that is a valid password. You can assume that if there are K spaces in string S then there are exactly K + 1 words.

   For example, given "test 5 a0A pass007 ?xy1", there are five words and three of them are valid passwords: "5", "a0A" and "pass007". Thus the longest password is "pass007" and its length is 7. Note that neither "test" nor "?xy1" is a valid password, because "?" is not an alphanumerical character and "test" contains an even number of digits (zero).

   Write a function:

   def solution(S)

   that, given a non-empty string S consisting of N characters, returns the length of the longest word from the string that is a valid password. If there is no such word, your function should return −1.

   For example, given S = "test 5 a0A pass007 ?xy1", your function should return 7, as explained above.

   Assume that:

   N is an integer within the range [1..200];
   string S consists only of printable ASCII characters and spaces.

2. You are given an array A consisting of the integers −1, 0 and 1. A slice of that array is any pair of integers (P, Q) such that 0 ≤ P ≤ Q < N. Your task is to find the longest slice of A whose elements yield a non-negative sum.

   Write a function:

   def solution(A)

   that, given an array A of length N, consisting only of the values −1, 0, 1, returns the length of the longest slice of A that yields a non-negative sum. If there's no such slice, your function should return 0.

   For example, given A = [−1, −1, 1, −1, 1, 0, 1, −1, −1], your function should return 7, as the slice starting at the second position and ending at the eighth is the longest slice with a non-negative sum.

   For another example, given A = [1, 1, −1, −1, −1, −1, −1, 1, 1] your function should return 4: both the first four elements and the last four elements of array A are longest valid slices.

   Write an efficient algorithm for the following assumptions:

   N is an integer within the range [2..100,000];
   each element of array A is an integer within the range [−1..1].