Quiz- Inventing Tests

Total points 4

1.
Question 1 (1 pt)
In this quiz, we aren't asking you to solve problems. Instead, you are given some problems and their incorrect solutions, and your task is to find tests where these solutions actually work incorrectly — by returning incorrect answer, working for too long, crashing, taking too much memory — anything that would give a verdict other than "Accepted". Consider that the time limit is 2 seconds, and the memory limit is 256 megabytes for all of the following problems.

This exercise will help you imagine tests for your own code. And it's actually very similar to "challenges" on many competitive programming platforms (e.g. Topcoder, Codeforces) — where after solving a problem you could earn additional score by breaking other's solutions to it. We hope you'll like it!

Next follows the first problem and the incorrect solution to it.

You are given a non-empty list of integers, and you need to find the maximum value among them. The length of the list is not greater than 100, and the absolute value of each element is not greater than 1000.

```
def solve(a):
  max = 0
  for x in a:
    if x > max:
      max = x
  return max
```

Implement a function called **getTest**. It should return a list on which the **solve** function works incorrectly. Note that the returned list must fit the restrictions in the statement.

The function is to be implemented in Python 3, but if you don't know this language, it's no problem — the sample code should give you the idea of how to do what you need.

Example

```
def getTest():
  return [2,3] # a value
```

1 point

2.
Question 2 (1 pt)
The second problem.

You are given a non-empty list of integers. Find the maximum possible value which could be obtained as the sum of no more than two elements from the list. You can't take the same element twice, but you could take *two distinct elements which are equal by value. The length of the list is not greater than 100, each element is positive and not greater than 1000.*

```
def solve(a):
  max = 0
  for i in range(len(a)):
    for j in range(len(a)):
      if i != j and a[i] + a[j] > max:
        max = a[i] + a[j]
  return max
```

The task is the same — return a list which breaks the solution above.

1 point

3.

Question 3 (1 pt)

Given an integer N, find how many pairs of distinct integers from 0 to N - 1 sum up to an even value. (1, 2) and (2, 1) are the same pair. N is positive, and doesn't exceed 100000.

```
def solve(n):
  ans = 0
  for i in range(n):
    for j in range(n):
      if i < j and (i + j) % 2 == 0:
        ans += 1
  return ans
```

Again, challenge this solution!

1 point

4.

Question 4 (1 pt)

You are given a string of lowercase English letters. First, remove all occurrences of its first letter, e.g. "abacaba" -> "bcb". Then, return the longest prefix of the remaining string which doesn't contain two different letters. The initial string is at least 5 and no more than 100000 characters long.

```
def solve(s):
  toDelete = s[0]
  others = ""
  for c in s:
    if c != toDelete:
      others += c
  s = others

  #prefix of length 1 surely doesn't contain different letters
  prefix = s[0]
  for i in range(1, len(s)):
    #each letter should be the same as the first
    if s[i] == prefix[0]:
      prefix += s[i]
    else:
      break
  return prefix
```