Zonal Informatics Olympiad, 2021

Instructions to candidates

1. The question paper carries 80 marks, broken up into four problems of 20 marks each. Each problem has three Test Cases. If you solve all three Test Cases correctly, you get 20 marks for that problem. Otherwise, you get 5 marks for each Test Case that you solve correctly.

2. All the $4 \times 3 = 12$ Test Cases appear as separate Questions in the right panel ("Question Palette").

   The first three Questions correspond to the three Test Cases of Problem 1, the next three correspond to the three Test Cases of Problem 2 and so on.

   A question icon turning green in the Question Palette, does not mean that it is correct. It just denotes that you have attempted it. All the questions will be evaluated later.

3. Attempt all questions. There are no optional questions.

4. There are no negative marks.

5. All expected answers are integers. Type in only the integer. So, if your answer is "162", enter only "162". Not "0162", or "162.0", etc.

6. Remember to save each answer. Only your final saved answers will be considered.

7. Near the top-right corner, you should be able to see a calculator icon. Clicking it pops up a calculator which you may use.

Zonal Informatics Olympiad, 2021

Questions

1. A k-bounded list of length n is a sequence $[x_1, x_2, \ldots, x_n]$ where each x_j is an integer between 0 and k, inclusive. A good list is one that does not contain three consecutive values $0, 1, 0$. For example, for $n = 4$ and $k = 2$, $[2, 0, 0, 1]$ and $[1, 0, 0, 1]$ are good lists while $[0, 1, 0, 2]$ and $[0, 0, 1, 0]$ are not.

   Let good(n, k) be the number of good k-bounded lists of length n. For instance, the good 1-bounded lists of length 2 are $\{[0, 0], [0, 1], [1, 0], [1, 1]\}$, so good$(2, 1) = 4$. Similarly, you can check that good$(4, 1) = 12$ and good$(3, 2) = 26$.

   Your task is to compute good(n, k) for the following values of n and k.

   (a) good$(7, 1)$

   (b) good$(7, 3)$

   (c) good$(20, 1)$

2. Given a list $[x_1, x_2, \ldots, x_n]$ of integers, a prefix of length j, for $1 \leq j \leq n$, is the list $[x_1, x_2, \ldots, x_j]$, and the corresponding prefix sum is $x_1 + x_2 + \cdots + x_j$. The integers in the list may be negative, so a prefix sum could be negative.

   A list of length n has n prefixes and n corresponding prefix sums. Our goal is to ensure that no prefix sum is negative. To achieve this, we can flip the signs of some of the negative values in the list. For instance, one way to ensure that all prefix sums are non-negative is to flip the sign of each negative number, making all values non-negative.

   Our goal is to identify the minimum number of values in the list whose sign needs to be flipped so that no prefix sum is negative. For example, if the list is $[3, -3, -2, -1, 0]$, we can flip the sign of just one value, -2, to get $[3, -3, 2, -1, 0]$. The prefix sums of this new list are $(3, 0, 2, 1, 1)$, which are all non-negative.

   For each of the following lists, find the minimum number of values whose signs need to be flipped to ensure that all prefix-sums are non-negative.

   (a) $[3, -2, 3, -1, -2, -2, -4]$

   (b) $[-15, -12, -10, -13, -2, -3, -17, -19, -5, -9]$

   (c) $[-12, -2, -16, -19, -9, -3, -7, -11, -17, -3, -15, -10, -10, -15, -8]$

3. We call an integer solid if it is positive and we do not need to use the digit 0 to write it down in base 10. For example, 115 is solid, but 10 and 1205 are not. The 10^{th} solid integer is 11 because the first 10 solid integers are $1, 2, 3, 4, 5, 6, 7, 8, 9, 11$. For each of the following values of n, you have to compute the n^{th} solid integer.

   (a) $n = 100$

   (b) $n = 2000$

   (c) $n = 100000$

4. We are given two positive integers x and y. We want to update x in a sequence of steps to make it equal to y. A single update to x consists of one of the following steps, where k is a fixed positive integer.

   - Multiply x by k
   - Subtract 1 from x

   For instance, if $x = 2$, $y = 7$ and $k = 2$, we can convert x to y in 3 steps, as follows.

   - Multiply x by k to get 4
   - Multiply x by k to get 8
   - Subtract 1 from x to get 7

   Let $\text{convert}(x, y, k)$ be the minimum number of steps needed to convert x to y using k. We can check that $\text{convert}(2, 7, 2) = 3$, because we cannot do better than the steps described above.

   Compute $\text{convert}(x, y, k)$ for the following values of x, y and k:

   (a) $\text{convert}(3, 10, 2)$

   (b) $\text{convert}(4, 92, 3)$

   (c) $\text{convert}(11, 104250, 2)$
