Hi All,

Here is our second assignment of the series.

I have shortened the problem statement in order to make sure that you guys can finish this in minimum time.

However, one has to submit the earlier assignment along with this one. So make sure you get rid of backlogs.

**Concept:**

A binary gap within a positive integer N is any maximal sequence of consecutive zeros that is surrounded by

ones at both ends in the binary representation of N.

For example, number 9 has binary representation 1001 and contains a binary gap of length 2.

The number 529 has binary representation 1000010001 and contains two binary gaps: one of length 4 and one of length 3.

The number 20 has binary representation 10100 and contains one binary gap of length 1.

The number 15 has binary representation 1111 and has no binary gaps. The number 32 has binary representation 100000 and has no binary gaps.

**Problem Statement**

Write a function:

int solution(int N);

such that, given a positive integer N, returns the length of its longest binary gap.

The function should return 0 if N doesn't contain a binary gap.

For example, given N = 1041 the function should return 5, because N has binary representation 10000010001

and so its longest binary gap is of length 5.

Given N = 32 the function should return 0, because N has binary representation '100000' and thus no binary gaps.

Assume that:

N is an integer within the range [1..2,147,483,647].

Instructions:

- First two days are kept for understanding the problem. Hence utilize this time properly and then only start working on actual problem.

- It is expected to understand the design and flow of problem solution, hence draw flow charts and dummy codes before coding.

- Most important ASK QUESTIONS to Me, Amol, Anil anyone with whom you feel comforatable

- Deadline for this assignment is definitely shorter than earlier one.

- Don't forgot to clear your backlogs. You have to submit the first assignment.

If anything is unclear, please come and ask.

Submit assignment within two days.