



CodeCheck Report: trainingUVE6EJ-GYF

[Check out Codility training tasks](#)

Test Name:

- Summary
- Timeline
- AI Assistant Transcript

Tasks summary

Task	Time spent	Score
BinaryGap	31 min	100%

Total score



Tasks Details

Easy	1. BinaryGap	Task Score	Correctness	Performance
	Find longest sequence of zeros in binary representation of an integer.			
		100%	100%	Not assessed

Task description

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.

Write a function:

```
int solution(int N);
```

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

Solution

Programming language used:	C++
Total time used:	31 minutes
Effective time used:	31 minutes
Notes:	not defined yet

Task timeline

08:41:37

09:12:25

Code: 09:12:25 UTC, cpp, [show code in pop-up](#)

final, score: 100

1

// you can use includes, for example:

2

// #include <algorithm>

should return 0, because N has binary representation '100000' and thus no binary gaps.

Write an **efficient** algorithm for the following assumptions:

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

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
3
4 // you can write to stdout for debugging purposes,
5 // cout << "this is a debug message" << endl;
6
7 int solution(int N) {
8     // Implement your solution here
9     int ans=0;
10    int count=0; // to count total bits in binary r
11    int start=-1; // to show the initial one
12
13    while(N)
14    {
15        int curr=N & 1; // get the last bit(0/1) o
16
17        if(curr==1)
18        {
19            // current bit is one
20            if(start!=-1)
21            {
22                ans = max(ans, count-start-1);
23            }
24            start=count;
25        }
26        // increase the count
27        count++;
28        // reduce the N
29        N=N/2;
30    }
31    return ans;
32
33 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Example tests	
▶ example1	✓ OK
example test n=1041=10000010001_2	
▶ example2	✓ OK
example test n=15=1111_2	
▶ example3	✓ OK
example test n=32=100000_2	
Correctness tests	
▶ extremes	✓ OK
n=1, n=5=101_2 and n=2147483647=2**31-1	
▶ trailing_zeroes	✓ OK
n=6=110_2 and n=328=101001000_2	
▶ power_of_2	✓ OK
n=5=101_2, n=16=2**4 and n=1024=2**10	
▶ simple1	✓ OK
n=9=1001_2 and n=11=1011_2	
▶ simple2	✓ OK
n=19=10011 and n=42=101010_2	
▶ simple3	✓ OK
n=1162=10010001010_2 and n=5=101_2	
▶ medium1	✓ OK
n=51712=110010100000000_2 and	

n=20=10100_2		
▶ medium2	✓ OK	n=561892=10001001001011100100_2 and n=9=1001_2
▶ medium3	✓ OK	n=66561=10000010000000001_2
▶ large1	✓ OK	n=6291457=11000000000000000000 001_2
▶ large2	✓ OK	n=74901729=1000111011011101000 11100001
▶ large3	✓ OK	n=805306373=1100000000000000000 00000000101_2
▶ large4	✓ OK	n=1376796946=10100100001000001 00000100010010_2
▶ large5	✓ OK	n=1073741825=1000000000000000000 0000000000001_2
▶ large6	✓ OK	n=1610612737=1100000000000000000 0000000000001_2