Calculate XOR from 1 to n.

Difficulty Level : Medium ● Last Updated : 30 Aug, 2021

Given a number n, the task is to find the XOR from 1 to n.

Examples :

```
Input : n = 6
Output : 7
// 1 ^ 2 ^ 3 ^ 4 ^ 5 ^ 6  = 7

Input : n = 7
Output : 0
// 1 ^ 2 ^ 3 ^ 4 ^ 5 ^ 6 ^ 7 = 0
```

Recommended: Please try your approach on ***{IDE}*** first, before moving on to the solution.

Method 1 (Naive Approach):

1- Initialize result as 0.

1- Traverse all numbers from 1 to n.

2- Do XOR of numbers one by one with result.

3- At the end, return result.

Method 2 (Efficient method) :

1- Find the remainder of n by moduling it with 4.

2- If rem = 0, then xor will be same as n.

3- If rem = 1, then xor will be 1.

4- If rem = 2, then xor will be n+1.

5- If rem = 3 ,then xor will be 0.

C++

```
// C++ program to find XOR of numbers
// from 1 to n.
#include<bits/stdc++.h>
using namespace std;

// Method to calculate xor
int computeXOR(int n)
{

  // If n is a multiple of 4
  if (n % 4 == 0)
    return n;

  // If n%4 gives remainder 1
  if (n % 4 == 1)
    return 1;

  // If n%4 gives remainder 2
  if (n % 4 == 2)
    return n + 1;

  // If n%4 gives remainder 3
  return 0;
}

// Driver method
int main()
{
  int n = 5;
  cout<<computeXOR(n);
}
```

```
// This code is contributed by rutvik_56.
```

Java

```
// Java program to find XOR of numbers
// from 1 to n.

class GFG
{
    // Method to calculate xor
    static int computeXOR(int n)
    {
        // If n is a multiple of 4
        if (n % 4 == 0)
            return n;

        // If n%4 gives remainder 1
        if (n % 4 == 1)
            return 1;

        // If n%4 gives remainder 2
        if (n % 4 == 2)
            return n + 1;

        // If n%4 gives remainder 3
        return 0;
    }

    // Driver method
    public static void main (String[] args)
    {
         int n = 5;
         System.out.println(computeXOR(n));
    }
}
```

Python 3

```
# Python 3 Program to find
# XOR of numbers from 1 to n.
# Function to calculate xor
def computeXOR(n) :

    # Modulus operator are expensive
    # on most of the computers. n & 3
    # will be equivalent to n % 4.

    # if n is multiple of 4
    if n % 4 == 0 :
        return n

    # If n % 4 gives remainder 1
    if n % 4 == 1 :
        return 1

    # If n%4 gives remainder 2
    if n % 4 == 2 :
        return n + 1

    # If n%4 gives remainder 3
    return 0

# Driver Code
if __name__ == "__main__" :

    n = 5

    # function calling
    print(computeXOR(n))

# This code is contributed by ANKITRAI1
```

C#

```
// C# program to find XOR
// of numbers from 1 to n.
using System;

class GFG
{

    // Method to calculate xor
    static int computeXOR(int n)
    {
        // If n is a multiple of 4
        if (n % 4 == 0)
            return n;

        // If n%4 gives remainder 1
        if (n % 4 == 1)
            return 1;

        // If n%4 gives remainder 2
        if (n % 4 == 2)
            return n + 1;

        // If n%4 gives remainder 3
        return 0;
    }

    // Driver Code
    static public void Main ()
    {
        int n = 5;
        Console.WriteLine(computeXOR(n));
    }
}

// This code is contributed by ajit
```

PHP

```
<?php
// PHP program to find XOR
// of numbers from 1 to n.

// Function to calculate xor
function computeXOR($n)
{
    // Modulus operator are expensive
    // on most of the computers. n & 3
    // will be equivalent to n % 4.

    switch($n & 3) // n % 4
    {
    // if n is multiple of 4
    case 0: return $n;
    // If n % 4 gives remainder 1
    case 1: return 1;

    // If n % 4 gives remainder 2
    case 2: return $n + 1;

    // If n % 4 gives remainder 3
    case 3: return 0;
    }
}

// Driver code
$n = 5;
echo computeXOR($n);

// This code is contributed by aj_36
?>
```

Javascript

```
<script>

// JavaScript program to find XOR of numbers
// from 1 to n.

// Function to calculate xor
function computeXOR(n)
{
    // Modulus operator are expensive on most of the
    // computers. n & 3 will be equivalent to n % 4.

    // if n is multiple of 4
    if(n % 4 == 0)
        return n;
    // If n % 4 gives remainder 1
    if(n % 4 == 1)
        return 1;
    // If n % 4 gives remainder 2
    if(n % 4 == 2)
        return n + 1;
    // If n % 4 gives remainder 3
    if(n % 4 == 3)
        return 0;
}

// Driver code

    // your code goes here
    let n = 5;
    document.write(computeXOR(n));

// This code is constributed by Surbhi Tyagi.

</script>
```

Output :

```
1
```

How does this work?

When we do XOR of numbers, we get 0 as XOR value just before a multiple of 4. This keeps repeating before every multiple of 4.

```
Number Binary-Repr  XOR-from-1-to-n
```

```
1         1           [0001]
2        10           [0011]
3        11           [0000]  <----- We get a 0
4       100           [0100]  <----- Equals to n
5       101           [0001]
6       110           [0111]
7       111           [0000]  <----- We get 0
8      1000           [1000]  <----- Equals to n
9      1001           [0001]
10     1010           [1011]
11     1011           [0000] <------ We get 0
12     1100           [1100] <------ Equals to n
```

This article is contributed by Sahil Chhabra. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.
Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Like 104

Next

Find XOR of two number without using XOR operator

RECOMMENDED ARTICLES

Page : 1 2 3

01 Calculate Bitwise OR of two integers from their given Bitwise AND and Bitwise XOR values
02, Dec 20

02 Queries to calculate maximum Bitwise XOR of X with any array element not exceeding M

05 Count of integers K in range [0, N] such that (K XOR K+1) equals (K+2 XOR K+3)
06, Jul 21

06 Bitwise XOR of same indexed array elements after rearranging an array to make

23, Mar 21

03 **Given a set, find XOR of the XOR's of all subsets.**
27, Dec 15

04 **Maximize count of pairs whose bitwise XOR is even by replacing such pairs with their Bitwise XOR**
22, Feb 21

XOR of same indexed elements of two arrays equal
05, Aug 21

07 **Find XOR of two number without using XOR operator**
21, Oct 15

08 **Choose X such that (A xor X) + (B xor X) is minimized**
03, May 19

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Medium

Easy Normal Medium Hard Expert

Improved By : Sameer Chaudhari 1, jit_t, ankthon, surbhityagi15, rutvik_56, tejasvigupta723, simmytarika5

Article Tags : Bitwise-XOR, Bit Magic

Practice Tags : Bit Magic

Improve Article Report Issue