# Check if a number is multiple of 9 using bitwise operators

Given a number n, write a function that returns true if n is divisible by 9, else false. The most simple way to check for n's divisibility by 9 is to do n%9.
Another method is to sum the digits of n. If sum of digits is multiple of 9, then n is multiple of 9.
The above methods are not bitwise operators based methods and require use of % and /.
The bitwise operators are generally faster than modulo and division operators. Following is a bitwise operator based method to check divisibility by 9.

```
#include<iostream>
using namespace std;

// Bitwise operator based function to check divisibility by 9
bool isDivBy9(int n)
{
    // Base cases
    if (n == 0 || n == 9)
        return true;
    if (n < 9)
        return false;

    // If n is greater than 9, then recur for [floor(n/9) - n%8]
    return isDivBy9((int)(n>>3) - (int)(n&7));
}

// Driver program to test above function
int main()
{
    // Let us print all multiples of 9 from 0 to 100
    // using above method
    for (int i = 0; i < 100; i++)
       if (isDivBy9(i))
```

Google™ Custom Search

GeeksforGeeks

53,527 people like GeeksforGeeks.



open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        cout << i << " ";
    return 0;
}
```

Output:

```
0 9 18 27 36 45 54 63 72 81 90 99
```

**How does this work?**
*n/9* can be written in terms of *n/8* using the following simple formula.

```
n/9 = n/8 - n/72
```

Since we need to use bitwise operators, we get the value of *floor(n/8)* using *n>>3* and get value of *n%8* using *n&7*. We need to write above expression in terms of *floor(n/8)* and *n%8*.
*n/8* is equal to *"floor(n/8) + (n%8)/8"*. Let us write the above expression in terms of *floor(n/8)* and *n%8*

```
n/9 = floor(n/8) + (n%8)/8 - [floor(n/8) + (n%8)/8]/9
n/9 = floor(n/8) - [floor(n/8) - 9(n%8)/8 + (n%8)/8]/9
n/9 = floor(n/8) - [floor(n/8) - n%8]/9
```

From above equation, *n* is a multiple of *9* only if the expression *floor(n/8) – [floor(n/8) - n%8]/9* is an integer. This expression can only be an integer if the sub-expression *[floor(n/8) - n%8]/9* is an integer. The subexpression can only be an integer if *[floor(n/8) - n%8]* is a multiple of *9*. So the problem reduces to a smaller value which can be written in terms of bitwise operators.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

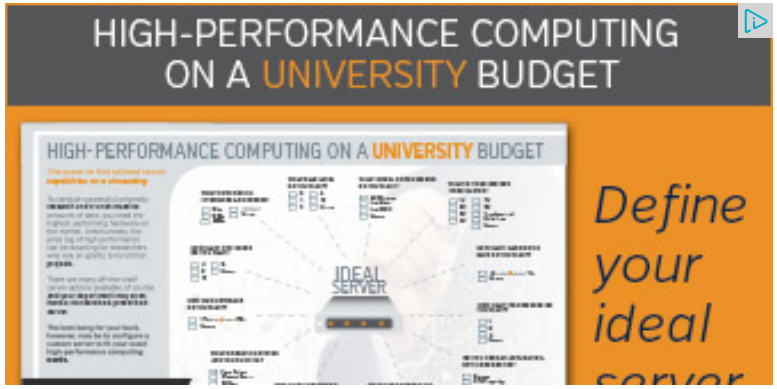
Geometric Algorithms

Sorry, spreadsheets.
Agile's got a new best friend.
Plan & track with JIRA Agile »
Atlassian JIRA Agile

## Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding “extern” keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com






open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com


## Related Tpoics:

- How to swap two numbers without using a temporary variable?
- Divide and Conquer | Set 4 (Karatsuba algorithm for fast multiplication)
- Find position of the only set bit
- Swap all odd and even bits
- Add two bit strings
- Write your own strcmp that ignores cases
- Binary representation of a given number
- Find the element that appears once

20 Tweet 3 1

**Writing code in comment?** Please use **ideone.com** and share the link here.

**8 Comments** GeeksforGeeks

Sort by Newest

Join the discussion…

**Vrushali** • 13 days ago

Hi All,

There is one more solution to it.A number is divisble by 9 if the sum of its digits

• Reply • Share ›

Market research that's fast and accurate.

Get $75 off

Google consumer surveys

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

**Lohith Ravi** • a month ago

dude you are using subtration, i would rather subtract 9, until the original numb

```
isDivisbleby9(int n)
{
if(n==0) return true;
else if(n < 9) return false;

return isDivisbleby9(n-9);
}
```

 • Reply • Share ›

**Bhagat Vishal** Lohith Ravi • 8 days ago

dude you just used some extra stack . problem talks about bitwise ope to this approach , you need to find the solution using bitwise operators.

 • Reply • Share ›

**inquisitive** • 2 months ago

Are we expected to come up with this solution during an interview? Even if I dc apparent.

2  • Reply • Share ›

**Mayank Rajani** • 2 months ago

You are eventually using minus(-) which is not a bitwise operator.Not a very ef I would use n%9 instead. Pls correct me if I m wrong.

4  • Reply • Share ›

**Abhishek Kumar** • 2 months ago

gud one...!!!!


 Subscribe

## Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 31 minutes ago

**Aman** Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 1 hour ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

**Sanjay Agarwal** bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

**GOPI GOPINATH** @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

**newCoder3006** If the array contains negative numbers also. We...

Find subarray with given sum · 2 hours ago

AdChoices

▶ C++ Code

▶ Java Source Code

▶ Programming C++

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

• Reply • Share ›

**sachi059** • 2 months ago

Nice idea

• Reply • Share ›

**Koushik** • 2 months ago

Good Solution.....

• Reply • Share ›


Subscribe Add Disqus to your site

AdChoices

▶ C++ Example

▶ Numbers Number

▶ Prime Number

AdChoices

▶ C++ Example

▶ Numbers Number

▶ Java Array





open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com