

## Add two numbers without using arithmetic operators

Write a function Add() that returns sum of two integers. The function should not use any of the arithmetic operators (+, ++, -, -, .. etc).

Sum of two bits can be obtained by performing XOR (^) of the two bits. Carry bit can be obtained by performing AND (&) of two bits.

Above is simple **Half Adder** logic that can be used to add 2 single bits. We can extend this logic for integers. If x and y don't have set bits at same position(s), then bitwise XOR (^) of x and y gives the sum of x and y. To incorporate common set bits also, bitwise AND (&) is used. Bitwise AND of x and y gives all carry bits. We calculate (x & y) << 1 and add it to x ^ y to get the required result.

```
#include<stdio.h>
```

```
int Add(int x, int y)
{
    // Iterate till there is no carry
    while (y != 0)
    {
        // carry now contains common set bits of x and y
        int carry = x & y;

        // Sum of bits of x and y where at least one of the bits is not set
        x = x ^ y;

        // Carry is shifted by one so that adding it to x gives the required sum
        y = carry << 1;
    }
    return x;
}
```

```
int main()
```



```
{
    printf("%d", Add(15, 32));
    return 0;
}
```

Following is recursive implementation for the same approach.

```
int Add(int x, int y)
{
    if (y == 0)
        return x;
    else
        return Add(x ^ y, (x & y) << 1);
}
```

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



## Related Tpoics:

- Check if a number is multiple of 9 using bitwise operators
- How to swap two numbers without using a temporary variable?
- Divide and Conquer | Set 4 (Karatsuba algorithm for fast multiplication)



## 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!

Structure Member Alignment, Padding and Data Packing

Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

Sorted Linked List to Balanced BST

- 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



45



Tweet

0



8

Writing code in comment? Please use [ideone.com](https://ideone.com) and share the link here.

Shouldn't  
you expect  
a cloud with:

**ONE-CLICK  
DEPLOYMENTS**

Experience the  
Managed Cloud  
Difference

TRY TODAY ►

 **rackspace**  
the open cloud company



## Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 23 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 · 1 hour ago

AdChoices

[► C++ Code](#)

[► Programming C++](#)


[► Negative Numbers](#)

AdChoices 

► [Convert Numbers](#)

► [Logic Numbers](#)

► [5 Numbers](#)

AdChoices 

► [Addition Numbers](#)

► [Numbers Number](#)

► [Adding Numbers](#)

@geeksforgeeks, **Some rights reserved**

**Contact Us!**

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team