Add 1 to a given number

Write a program to add one to a given number. You are not allowed to use operators like '+', '-', '*', '/', '++', '--' ...etc.

Examples:
Input: 12
Output: 13

Input: 6
Output: 7

Yes, you guessed it right, we can use bitwise operators to achieve this. Following are different methods to achieve same using bitwise operators.

Method 1
To add 1 to a number x (say 0011000111), we need to flip all the bits after the rightmost 0 bit (we get 0011000000). Finally, flip the rightmost 0 bit also (we get 0011001000) and we are done.

```
#include<stdio.h>

int addOne(int x)
{
  int m = 1;

  /* Flip all the set bits until we find a 0 */
  while( x & m )
  {
    x = x^m;
    m <<= 1;
  }
```


```
  /* flip the rightmost 0 bit */
  x = x^m;
  return x;
}

/* Driver program to test above functions*/
int main()
{
  printf("%d", addOne(13));
  getchar();
  return 0;
}
```

Method 2

We know that the negative number is represented in 2′s complement form on most of the architectures. We have the following lemma hold for 2′s complement representation of signed numbers.

Say, x is numerical value of a number, then

~x = -(x+1) [~ is for bitwise complement]

(x + 1) is due to addition of 1 in 2′s complement conversion

To get (x + 1) apply negation once again. So, the final expression becomes (-(~x)).

```
int addOne(int x)
{
   return (-(~x));
}

/* Driver program to test above functions*/
int main()
{
  printf("%d", addOne(13));
  getchar();
  return 0;
}
```

Example, assume the machine word length is one *nibble* for simplicity.
And x = 2 (0010),
~x = ~2 = 1101 (13 numerical)
-~x = -1101
Interpreting bits 1101 in 2′s complement form yields numerical value as -(2^4 – 13) = -3. Applying

Geometric Algorithms

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

'-' on the result leaves 3. Same analogy holds for decrement. See this comment for implementation of decrement.
Note that this method works only if the numbers are stored in 2's complement form.

Thanks to Venki for suggesting this method.

Please write comments if you find the above code/algorithm incorrect, or find better ways to solve the same problem

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)
- 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

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

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 22 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