# **GeeksforGeeks**

A computer science portal for geeks

Login

Home	Algorithms	DS	GATE In	nterview Corner	Q&A	С	C++	Java	Books	Contribute	Ask a Q	About
Array	Bit Magic	C/C+	+ Article	s GFacts	Linked Li	st	MCQ	Misc	Outpu	t String	Tree	Graph

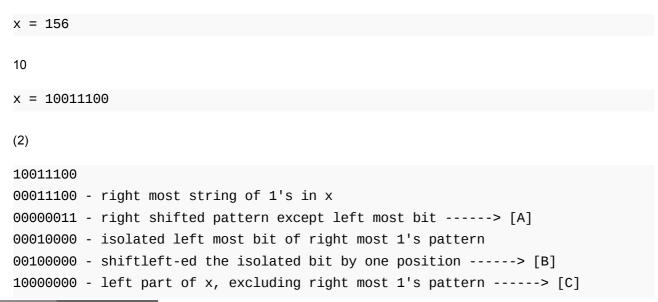
# Next higher number with same number of set bits

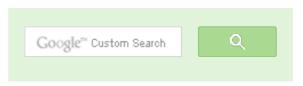
Given a number x, find next number with same number of 1 bits in it's binary representation.

For example, consider x = 12, whose binary representation is 1100 (excluding leading zeros on 32 bit machine). It contains two logic 1 bits. The next higher number with two logic 1 bits is 17 (10001<sub>2</sub>).

### Algorithm:

When we observe the binary sequence from 0 to  $2^{n} - 1$  (n is # of bits), right most bits (least significant) vary rapidly than left most bits. The idea is to find right most string of 1's in x, and shift the pattern to right extreme, except the left most bit in the pattern. Shift the left most bit in the pattern (omitted bit) to left part of x by one position. An example makes it more clear,







53,526 people like GeeksforGeeks.











### Interview Experiences

Advanced Data Structures

Dynamic Programming

**Greedy Algorithms** 

Backtracking

Pattern Searching

Divide & Conquer

Mathematical Algorithms

Recursion

Geometric Algorithms

```
10100000 - add B and C (OR operation) -----> [D]
10100011 - add A and D which is required number 163
```

(10)

After practicing with few examples, it easy to understand. Use the below given program for generating more sets.

### **Program Design:**

We need to note few facts of binary numbers. The expression x & -x will isolate right most set bit in x (ensuring x will use 2's complement form for negative numbers). If we add the result to x, right most string of 1's in x will be reset, and the immediate '0' left to this pattern of 1's will be set, which is part [B] of above explanation. For example if x = 156, x & -x will result in 00000100, adding this result to x yields 10100000 (see part D). We left with the right shifting part of pattern of 1's (part A of above explanation).

There are different ways to achieve part A. Right shifting is essentially a division operation. What should be our divisor? Clearly, it should be multiple of 2 (avoids 0.5 error in right shifting), and it should shift the right most 1's pattern to right extreme. The expression (x & -x) will serve the purpose of divisor. An EX-OR operation between the number X and expression which is used to reset right most bits, will isolate the rightmost 1's pattern.

#### A Correction Factor:

Note that we are adding right most set bit to the bit pattern. The addition operation causes a shift in the bit positions. The weight of binary system is 2, one shift causes an increase by a factor of 2. Since the increased number (rightOnesPattern in the code) being used twice, the error propagates twice. The error needs to be corrected. A right shift by 2 positions will correct the result.

The popular name for this program is same number of one bits.

```
#include<iostream>
using namespace std;
typedef unsigned int uint t;
```



# 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

```
// this function returns next higher number with same number of set bi
uint t snoob(uint t x)
  uint t rightOne;
 uint t nextHigherOneBit;
 uint t rightOnesPattern;
  uint t next = 0;
  if(x)
    // right most set bit
    rightOne = x & -(signed)x;
    // reset the pattern and set next higher bit
    // left part of x will be here
    nextHigherOneBit = x + rightOne;
    // nextHigherOneBit is now part [D] of the above explanation.
    // isolate the pattern
    rightOnesPattern = x ^ nextHigherOneBit;
    // right adjust pattern
    rightOnesPattern = (rightOnesPattern)/rightOne;
    // correction factor
    rightOnesPattern >>= 2;
    // rightOnesPattern is now part [A] of the above explanation.
    // integrate new pattern (Add [D] and [A])
    next = nextHigherOneBit | rightOnesPattern;
  return next;
int main()
  int x = 156;
  cout<<"Next higher number with same number of set bits is "<<snoob(x</pre>
  getchar();
  return 0;
```

Custom market research at scale.

Get \$75 off

Google consumer surveys

}

**Usage:** Finding/Generating subsets.

### Variations:

1. Write a program to find a number immediately smaller than given, with same number of logic 1 bits? (Pretty simple)

2. How to count or generate the subsets available in the given set?

### References:

- 1. A nice presentation here.
- 2. Hackers Delight by Warren (An excellent and short book on various bit magic algorithms, a must for enthusiasts)
- 3. C A Reference Manual by Harbison and Steele (A good book on standard C, you can access code part of this post here).
- **Venki**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

# Computer Broken?

Melpouts.google.com/computer\_help

Connect With Live Experts Who Will Help You Fix It From Home Today.







## **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 [⊳

- ▶ Bits Byte
- ► Hex Bits
- ▶ Pattern Bits

AdChoices [>

► Numbers Number

## 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.

### 23 Comments

GeeksforGeeks

Sort by Newest ▼



Join the discussion...



Wellwisher • a month ago

Nice code in C#. Complexiety O(n)

http://onestopinterviewprep.bl...



ritu · 4 months ago

could you please explain in detail. i am still a beginer. plz



Omor J. Kocharee • 8 months ago

Why its too complicated?

- ▶ Bits Sets
- ► Math Number

AdChoices [>

- ► Number Counting
- ► Even Number
- ► Print a Number

loade cheek the http://tttt.enacehare.nergkann... for tory eacy approach.



Shradha Agrawal • 9 months ago

Nice one! I solved the question in similar manner.



Rick Giuly • 9 months ago

The procedure in Gayle McDowell&#039s book is as follows: First find the right i, set it to 1. Then consider the bits to the right of position I. Count how many 1 out and put N-1 high bits on in the least significant positions.

My comment here is a description of why I think this procedure actually works. problem X.

You know that at least one bit in X must have been flipped from 0 to 1 to make the most significant bit that was flipped high to produce the answer.

So, your answer has the form:

[left part] [bit i] [right part].

You know that somewhere there must be compensation (considering number high. This would mean that the number of bits in either the left or right part must

Now you are guaranteed that the number of bits on left will not decrease. The are too "course grained." Specifically, any bit in the [left part] flipped from high much that [bit il (and any other bits to the right of it) would not be enough to me

see more



Siddharth Rajpal • 10 months ago

There is another possible solution:

the immediate right to it to 0. Find the number of 1s to the left of it and add those number.

For example we have 100110:

The first 1 occurs on the second last bit. and the first 0 (after the second last t that 0 to 1 and the bit on the right of it to 0.

100110 -> (change the 0 to 1) 101110-> (change the bit on right of it to 0) 1010

now, count the number of 1s to the left of the 0 after 1010, we have only one 1

101010-> 101001.

Hence the answer is 101001.

```
1 ^ Reply · Share >
```



**Chen** → Siddharth Rajpal • 3 months ago

This is much more straightforward! Thanks!



aditya kumar • 10 months ago

first find out the pos of first rightmost '0' bit after(left of ) the rightmost set bit '1'

ex : num(42) = 101010 = = = > pos = 3; temp = 1 < > 1) = 101010 + 000100 - 0000code:

int next\_int\_with\_same\_set\_bit(int num) t=num&(-num); while(1) {t<>1); return ans;



Nishant Kumar • a year ago

Possible solution on GCC compilers without division operator



### Prateek • 2 years ago

```
/* Paste your code here (You may delete these lines if not writing co
void nextnum(unsigned int x){
    printf("%d\n",(n+(n^(n&(n-1))))|(n&(n-1)));
}
```



## **Agniswar** ⋅ 3 years ago

Hi,i solved in this way..though not so efficient but pretty simple one..

Link- http://ideone.com/NBMTG



```
pappu • 3 years ago

int nextNumber(int x)

{

int u = log(x & -x);

int y = x >> u;

int z = log(\simy & -(\simy));

int k = pow(2, z) - 1;

y = ((y >> 2) + 1) << (z + u);

return y + k;
```

```
}

    Reply • Share >
```



### AJ · 3 years ago

For immediate smaller number with same number of bits:

```
int next_lowest(int x)
{
    int removeones = (x + 1) & x;
    int isolate = removeones & ~(removeones - 1);
    int shifted = (removeones ^ isolate) | (isolate >> 1);
    int temp = (x + 1) & ~x;
    int factor = (shifted & ~(shifted - 1)) / (temp);
    int toadd = (temp - 1) * factor;

    return toadd|shifted;
}
```



insect → AJ · 10 months ago

Your program gives wrong output for 12.

```
/* Paste your code here (You may delete these lines if not wri
```





Manish Mishra • 3 years ago

To get the immediate lower number, just use this-

```
int nextsmallest(int n)
 return ~nextlargest(~n);
```



Imran Amjad • 3 years ago Hi GeeksforGeeks,

from the same logic to get the next higher number of same set bits, how i'll rev lower number? please give a brief explanation. Thanks



sutendra mirajkar · 3 years ago

```
#include<stdio.h>
#include<stdlib.h>
int main(int argc,char *argv[])
{
    int a,i,temp,bin,count=0,fcount=0,flag=0;
    if(argc != 2)
        printf("IMPROPER EXECUTION, PLEASE TYPE IN THE NUMBER AFTER ./a
    }
     a=atoi(argv[1]);
```

#### see more



naveen kolati • 3 years ago

```
#include<stdio.h>
int naveen(int );
void main()
{
  int p=0, k, n;
  printf("enter the number");
  scanf("%d",&n);
  k = naveen(n);
  while(k!=p)
     p=naveen(++n);
  printf("\n next number is %d",n);
  getch();
int naveen(int n)
  int count=0;
 while(n!=0)
    n=n&(n-1);
    count++;
  return count;
```





**Himanshu** ⋅ 3 years ago

Another method to find the lexicographical next permutation is given at followin

http://graphics.stanford.edu/~...



Preetam ⋅ 3 years ago

for 5 what is the next higher number? please write few more samples



naveen koati → Preetam · 3 years ago 6 is the next highest number after 5

6(110),5(101)



Venki → Preetam · 3 years ago

@Preetam, Use the sample program for generation of sets. You can s http://ideone.com/W1D5E.





Add Disgus to your site

@geeksforgeeks, Some rights reserved

Contact Us!

Powered by WordPress & MooTools, customized by geeksforgeeks team