Turn off the rightmost set bit

Write a C function that unsets the rightmost set bit of an integer.

Examples:

```
Input:  12 (00...01100)
Output: 8 (00...01000)

Input:  7 (00...00111)
Output: 6 (00...00110)
```

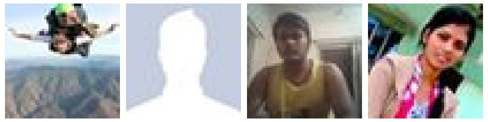
Let the input number be n. n-1 would have all the bits flipped after the rightmost set bit (including the set bit). So, doing n&(n-1) would give us the required result.

```
#include<stdio.h>

/* unsets the rightmost set bit of n and returns the result */
int fun(unsigned int n)
{
  return n&(n-1);
}

/* Driver program to test above function */
int main()
{
  int n = 7;
  printf("The number after unsetting the rightmost set bit %d", fun(n)

  getchar();
  return 0;
}
```

Google™ Custom Search

GeeksforGeeks

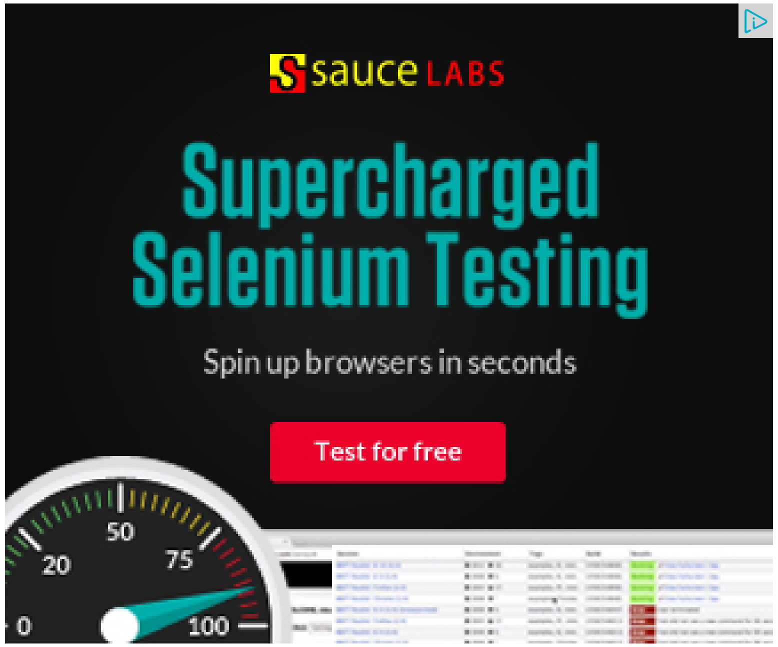
53,526 people like GeeksforGeeks.



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion

Geometric Algorithms

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



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

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

1 Tweet 0 0

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

18 Comments **GeeksforGeeks**

Join the discussion…

Castle Age • 3 months ago

Why do people love posting their codes instead of just commenting the post?

neelabhsingh • 6 months ago

turn of right most bit

```
int fun(int N)
{
int C=N&-N;
int B;
B=N^C;
return B;
}
```

Example: N=01101110
-N=10010010 (2's complement of N)
C=N&-N 00000010
B=N^C; (01101110)^(00000010)
B=01101100

Now you can see the rightmost bit is reset
If I am wrong then correct me. I waiting for the response.

pavansrinivas • 7 months ago

Code in JAVA..

```
        int c = 1;
        int i=0;
        while((c&x)<=0){
            i++;
            c = 1<<i; }="" x="x^(1&lt;&lt;i);" system.out.print(x);="'
```

1

Arindam Sanyal · a year ago

#include<stdio.h>
#include<conio.h>

void main(){
clrscr();
int a, i=0;
printf("ENTER A NUMBER TO SET THE RIGHTMOST BIT");.
scanf("%d",&a);
while((a|(1<<i))>a)
i++;
int k=a&~(1<<i);

printf("%d", k);

getch();
}

1

ARINDAM · a year ago

#include
#include

void main(){

705 Subscribe

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 20 minutes ago

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

Write a C program to Delete a Tree. · 59 minutes 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

clrscr();
int a,i=0;
printf("\nenter a number to turn off the rightmost set bit");
scanf("%d",&a);

while((a|(1<*a)*
i++;
int k=a&~(1<<i);

printf("%d",k);

getch();
}

vikas kumar • 2 years ago

#include

int fun(unsigned int n){
//base case when n=0
return n && ~(n&(n-1));
}

int main()
{
int n = 7;
printf("No after clear the rightmost set bit %d", fun(n));
return 0;
}

prakash • 2 years ago

#include

```
main()
{
int x,y=1;
scanf("%d",&x);
while(!(x & y))
{
y=y << 1;
}
x=x ^ y;
printf("%d\n",x);
return 0;
}
```

1

skulldude • 3 years ago

I think this will also do the required, though it is a bit more complex than the n&

res=x&~(x&-x)

aygul skulldude • a year ago

Actually if you normalize:
x&~(x&-x) = x&~x || x&~-x = x&~-x
which is the same thing with the given solution :)
because in two's complement -x = ~(x-1)

so: x&~-x = x&(x-1)

instead of writing 5 - 2 = 3 you write 5 + 2 - 6 +1 :)

Venki skulldude • 3 years ago

(x & -x) will reset all the bits from right most set bit (excluding right mos
most string of 0s preceded by 1, an exact power of 2).

~(x & -x) results as all left bits to 1 and right most set bit to 0, followed
right most set bit. :)

x & (~(x & -x)) - resets the rightmost set bit.

Good logic, but costly.

Where as the logic provided in post is bases on the fact that right mos
system.

shivam • 3 years ago

```
 int fun(unsigned int n)
{
  int temp= n & -n;
  return n^t;
}
```

shivam shivam • 3 years ago

sorry temp instead of t written there

casillas • 3 years ago

do a right shift and a left shift
n=n>>1;
n=n<<1;

santosh ➜ casillas • 3 years ago

It works for the numbers which had a set bit at last position.
Ex: 7 --> 0111 -- It works

But if last bit is "0" then it wont works..

Ex: 12 --> 1100
12>>1 --> 0110
now < 1100

so this logic is wrong...

correct one is n&(n-1)

Suresh • 3 years ago

```
 int unSetRightMostSetBit(int x)
{
     int m=1;
     while(!(x&m))
            m = m<<1;
     return x^m;
}

int main(void)
{
    int num;
    printf("Enter a number : ");
    scanf("%d",&num);
    printf("\nEntered Number : %d",num);
    printf("Result : %d",unSetRightMostSetBit(num));
}
```

Venki • 4 years ago

From the question, if we iterate successively till [n & (n-1)] becomes zero, it i
bits (1 s). However it is not efficient on highly pipelined machines. We can cou
complexity. For hint on the logarithmic algorithm see the following link, comme

http://math-puzzles-computing.blogspot.com/2010/06/bit-reversal_02.html

Sambasiva • 4 years ago

For input: 12, output: 8

GeeksforGeeks → Sambasiva • 4 years ago

Thanks for pointing this out. There was a typo in explanation. The prog
correct.

Subscribe Add Disqus to your site