Write a C program to find the parity of an unsigned integer

Parity: Parity of a number refers to whether it contains an odd or even number of 1-bits. The number has “odd parity”, if it contains odd number of 1-bits and is “even parity” if it contains even number of 1-bits.
Main idea of the below solution is – Loop while n is not 0 and in loop unset one of the set bits and invert parity.

```
Algorithm: getParity(n)
1. Initialize parity = 0
2. Loop while n != 0
      a. Invert parity
            parity = !parity
      b. Unset rightmost set bit
            n = n & (n-1)
3. return parity
```

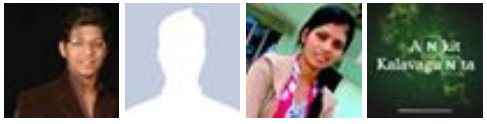
Example:

```
 Initialize: n = 13 (1101)   parity = 0

n = 13 & 12  = 12 (1100)   parity = 1
n = 12 & 11 = 8  (1000)   parity = 0
n = 8 & 7 = 0  (0000)    parity = 1
```

Program:

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

```
# define  bool int

/* Function to get parity of number n. It returns 1
   if n has odd parity, and returns 0 if n has even
   parity */
bool getParity(unsigned int n)
{
    bool parity = 0;
    while (n)
    {
        parity = !parity;
        n      = n & (n - 1);
    }
    return parity;
}

/* Driver program to test getParity() */
int main()
{
    unsigned int n = 7;
    printf("Parity of no %d = %s",  n,
           (getParity(n)? "odd": "even"));

    getchar();
    return 0;
}
```

Above solution can be optimized by using lookup table. Please refer to Bit Twiddle Hacks[1st reference] for details.

Time Complexity: The time taken by above algorithm is proportional to the number of bits set. Worst case complexity is O(Logn).

Uses: Parity is used in error detection and cryptography.

References:

http://graphics.stanford.edu/~seander/bithacks.html#ParityNaive – last checked on 30 May 2009.

Popular Posts

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.

10 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

Amit Kumar • 4 months ago

```
void main()

{

int value=7,i=1,count=0;
while (i){
if(value & i){ ++count;}
i=i<<1;
}
printf(" value: %d", count);
```


PingPong • 4 months ago

Hi, I don't understand the definition, which is given blow:

"The number has “odd parity”, if it contains odd number of 1-bits and is “even | bits."
And
" It returns 1if n has odd parity, and returns 0 if n has even parity"

Thus, shouldn't the example above return the result below instead:

Initialize: n = 13 (1101) parity should be 1 because it has 3 1-bits

n= 12 (1100) parity should 0 because it has 2 1-bits

What information am I missing here, please correct me?

dusty • 9 months ago

Another possible solution :

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 27 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

```
 int main()
{
    int a,n;
    a=0;
    scanf("%d",&n);
    while(n>0)
    {
        if(n&1)
            a=!a;
        n>>=1;
    }
    printf("%d",a);
}
```

6 ▲ ▼ • Reply • Share ›

Rohan • 9 months ago

How the time complexity is O(log n) .. please explain ...
we are doing n-1 on every iteration..!!! so it should be O(n) right ?

▲ ▼ • Reply • Share ›

Gopi Chan • a year ago

here by getting the value of k odd or even we can say that the number has odc

#include<stdio.h>
#include<math.h>
int parity(int n);.
int main()
{
int n, m;
n=26;

```
m=parity(n);
printf("%d", m);
}

int parity(int n).
{
int i, j,k;
k=0;
i=0;
while(n!=0)
```

see more

1 ▲ | ▼ • Reply • Share ›

Ritesh • a year ago

We can also use the gcc built-in function:
int __builtin_parity (unsigned int x)

Returns the parity of x, i.e. the number of 1-bits in x modulo 2.

1 ▲ | ▼ • Reply • Share ›

yaser fathi • 3 years ago

thanks for this program I want to C++ code for two-spiral problem and functior trainning

▲ | ▼ • Reply • Share ›

Lokesh • 4 years ago

Another optimized code:

```
 bool getParity(unsigned int x)
{
    x = ((x >> 1) & 0x55555555) + (x & 0x55555555);
    x = ((x >> 2) & 0x33333333) + (x & 0x33333333);
```

```
    x = ((x >> 4) & 0x0F0F0F0F) + (x & 0x0F0F0F0F);
    x = ((x >>  8 ) + x ;
    x = ((x >>  16 ) + x ;
    return (x&0x01);
}
```

Constant running time for any value. Worst Case Complexity O(1)

• Reply • Share ›

temp Lokesh • 3 years ago

Can you please explain what are these hex values? a link to a good so

Thanks

• Reply • Share ›

bala temp • 3 years ago

Unsigned int in the question takes 4 bytes.

0x55555555 - 01010101 01010101 01010101 01010101
0x33333333 - 00110011 00110011 00110011 00110011
0x0F0F0F0F - 00001111 00001111 00001111 00001111

What the first 3 steps basically does is that , it calculates the n
bits, 8 bits respectively.

The 4th step is actually not necessary, still it gives the correct a
omitted.

If you work out the first 3 steps , you will understand what the 4

• Reply • Share ›