

Functions | Pointers | Arrays | Strings

* Note: If there is only one non-repeating element in the array & all other elements are repeated twice, use the XOR of all the elements with zero(0) to get the unique element in Big O(n) time complexity.

What if there are two non-repeating elements?
Example: `int arr[] = {1, 2, 3, 4, 2, 1};`
→ right most set bit

Hamming Weight? `11 → 1011`

no. of 1's

no. of set bits → 191

`int hammingWeight(int n) {`

`5 → 0101 → 2`

`11 → 1011 → 3`

`} 0 → 0000 → 0`

`n = 11 → 1011`
`n & 1`
`0001`
`0001 → True → cnt++`
`1011 >> 1 = 101`
`& 001`
`001 → 1`
`10 >> 1`
`1`
`1 → 1`
`101 >> 1 = 10`
`1 >> 1 = 0`
`stop`
`TCO`

Given an integer value 'n', write a function to reverse the given integer.

`int reverseInteger(int n)`
`{`

`123 → 1×100 + 2×10 + 3`

`n = 123`

`o/p = 321`

`n = 120`

`o/p = 21`

`1000`
`int(3.96) + 1`
`2 + 1 = 4`
`123`
`12`
`1`
`0`
`1 2 3 4 → digits = 4`
`n/10`
`while n > 0`
`n = n/10`
`digits++`
`(int) log10(num)`
`decimal`

Baker's or Caterer's Partition Problem

`2xS + 1`
`11`
`S(n) + 1`

No. of Cuts	Max Pieces
0	1
1	2
2	4
3	7
4	11
5	16

Max Regions in a Plane with N straight lines.

`S(n) + 1`

No. of Straight Lines	Max Regions
<code>S(0) + 1</code>	1
<code>S(1) + 1</code>	2
<code>S(2) + 1</code>	4
3	7
4	11
5	16

Two numbers are given :→

`a, b → 15 60`

and you have to calculate their LCM.

Euclid's Algorithm of Repetitive Division

`45, 75 → GCD = 15`

`LCM × GCD`

`= a × b`

`45`
`75`
`45`
`30`
`45`
`30`
`15`
`30`
`15`
`15`
`15`
`0`

`int gcd(int a, int b) {`

`if (b == 0) {`

`return a;`

`}`

`return gcd(b, a/b);`

`}`

`45 : 75`
`= 45`

`a, b`
`15, 0`

`↓`

`a`

`45, 75`

`a, b`

`45, 30`

`30, 15`

`15, 0`

`15`