



< reverse bits

$\underline{\text{Main Page}} \rightarrow \underline{\text{Exercises}} \rightarrow \underline{\text{Homework 2}} \rightarrow \underline{\text{C}} \rightarrow \underline{\text{Solve an Exercise}}$



You are working on problem set: Homework 2 (Pause)

bits_to_flip ♥

Language/Type: C <u>bitwise operators bit</u>

algorithms bit manipulation

Related Links:

Wikipedia: Ones' complement

Write a function named bits_to_flip

that accepts as parameters two integers a and b and returns the number of bits that would need to be flipped from 0 to 1 or vice versa to turn a's binary representation into b's. For example, if the call is bits_to_flip(42, 108), the binary representation of 42 is 0101010 preceded by 25 zeros, and the binary representation of 108 is 1101100 preceded by 25 zeros. To convert the first into the second you would need to flip three bits. So your function should return 3.

You should solve this problem using bitwise operators and arithmetic. Do not use any collections or strings to solve the problem. You may assume that an int occupies 32 bits in memory.

.

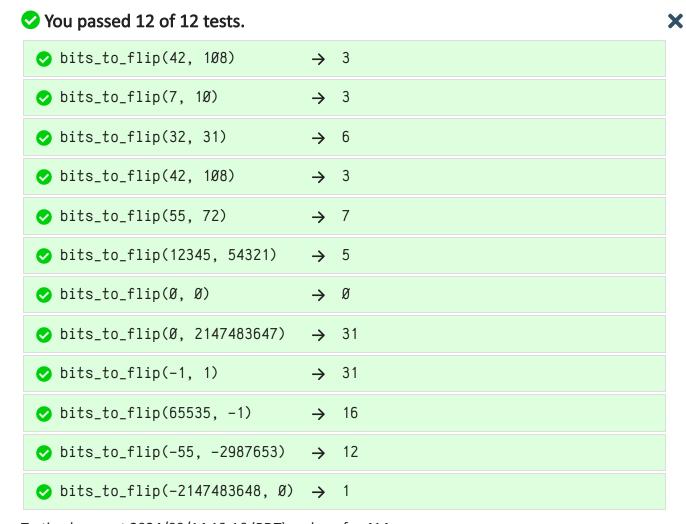
```
// bits_to_flip by Brandon Kmiec. Submitted for CSC152 September 15, 2024
// function to determine the number of bits to flip to turn a into b

int bits_to_flip(int a, int b) {
   int count = 0;
   int aXorB = a ^ b;

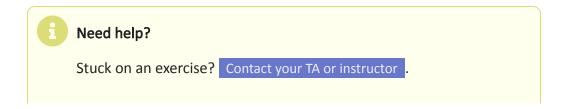
for(int i = 0; i < 32; i++) {
   if((aXorB & (1 << i)) > 0)
        count++;
```

Function: Write a C function as described, not a complete program.





Testing began at 2024/09/14 19:16 (PDT) and ran for 414 ms.



If something seems wrong with our site, please contact us.