FizzBuzz ★

Problem | Submissions | Leaderboard | Discussions

Consider the following problem:

> Write a short program that prints each number from 1 to 100 on a new line.
>
> For each multiple of 3, print "Fizz" instead of the number.
>
> For each multiple of 5, print "Buzz" instead of the number.
>
> For numbers which are multiples of both 3 and 5, print "FizzBuzz" instead of the number.

Write a solution (or reduce an existing one) so it has as few characters as possible.

```cpp
    /* FizzBuzz
     https://www.hackerrank.com/challenges/fizzbuzz/problem
    */

#include <bits/stdc++.h>
using namespace std;
 /*
 Basic idea- Iterating from 1 to 100, and checking for each
 number whether it is multiple of 3 and 5 both, or multiple of 3 only
 or multiple of 5 only, otherwise printing the number
 */
int main() {
     for(int i=1;i<=100;i++){
        if((i%3 == 0) && (i%5==0))
            cout<<"FizzBuzz\n";
        else if(i%3 == 0)
            cout<<"Fizz\n";
        else if(i%5 == 0)
            cout<<"Buzz\n";
        else
            cout<<i<<"\n";
     }
    return 0;
}
```

2427. Number of Common Factors

Easy | Topics | Companies | Hint

Given two positive integers `a` and `b`, return *the number of* ***common*** *factors of* `a` *and* `b`.

An integer `x` is a **common factor** of `a` and `b` if `x` divides both `a` and `b`.

Example 1:

```
Input: a = 12, b = 6
Output: 4
Explanation: The common factors of 12 and 6 are 1, 2, 3, 6.
```

Example 2:

```
Input: a = 25, b = 30
Output: 2
Explanation: The common factors of 25 and 30 are 1, 5.
```

Constraints:

- `1 <= a, b <= 1000`

```
/* Driver code-Header files
#include <bits/stdc++.h>
using namespace std;
*/
  /* https://leetcode.com/problems/number-of-common-factors/ */

   /*Basic idea is Common factors of both numbers will
   be upto smaller number only. Therfore iterated upto
   minm of these two, and checked whether the number is
   factor of both or not.*/
int commonFactors(int a, int b) {
        int count=0;
        for(int i=1;i<=min(a,b);i++)
        {
            if(a%i==0 && b%i==0)
                count++;
        }
        return count;
    }
/* Driver code
int main() {
    // taking input and calling function
    return 0;
}
*/
```

GCD of two numbers

School Accuracy: 51.03% Submissions: 58K+ Points: 0

Given two positive integers A and B, find GCD of A and B.

Example 1:

Input: A = 3, B = 6
Output: 3
Explanation: GCD of 3 and 6 is 3

Example 2:

Input: A = 1, B = 1
Output: 1
Explanation: GCD of 1 and 1 is 1

Your Task:
You don't need to read input or print anything. Complete the function **gcd()** which takes two positive integers as input parameters and returns an integer.

Expected Time Complexity: O(log(min(A, B)))
Expected Auxiliary Space: O(1)

Constraints:
$1 \leq A, B \leq 10^9$

```
https://www.geeksforgeeks.org/problems/gcd-of-two-numbers3459/1
*/
/* { Driver Code Starts
#include <bits/stdc++.h>
using namespace std;

// } Driver Code Ends
//User function Template for C++
*/
class Solution
{
    public:
    int gcd(int A, int B)
    {
        /* gcd of them will be smaller than or equal to minimum
        of these, then we just checked if the chosen number
        is dividing both of them or not
        */
        int n = A>B?B:A;
        int num;
        for(num = n; num>=1; num--) {
            if(A%num==0 && B%num==0)
                break;
        }
        return num;

    }
};

/*{ Driver Code Starts.

int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int A, B;
        cin >> A >> B;
        Solution ob;
        cout <<  ob.gcd(A, B) << "\n";
    }
    return 0;
}
 } Driver Code Ends
 */
```

231. Power of Two

Easy | Topics | Companies

Given an integer `n`, return *`true`* *if it is a power of two. Otherwise, return* *`false`*.

An integer `n` is a power of two, if there exists an integer `x` such that `n == 2^x`.

Example 1:

```
Input: n = 1
Output: true
Explanation: 2^0 = 1
```

Example 2:

```
Input: n = 16
Output: true
Explanation: 2^4 = 16
```

Example 3:

```
Input: n = 3
Output: false
```

Constraints:

- `-2^31 <= n <= 2^31 - 1`

```
https://leetcode.com/problems/power-of-two/
*/

/* Driver code
#include <bits/stdc++.h>
using namespace std;
*/
 /*

 If a number is perfect power of 2 then the number is in
form of 2 * 2 * 2 * ......* 2 * 1,
if we remove all factor of 2 from this number then at last
number is converted to 1.
so to check if a number is perfect power of we remove
all factor of 2 and at last if number is 1 then this is power
perfect of 2 otherwise not.
we can do this as */

class Solution {
public:
    bool isPowerOfTwo(int n) {
        if(n==0){
            return false;
        }
        while(n%2 == 0){
            n=n/2;
        }
        return n==1;
    }
};
/* Driver code
int main() {
    // taking value of n as input and calling function
    return 0;
}
```

Prime Number

Basic Accuracy: 22.2% Submissions: 205K+ Points: 1

For a given number **N** check if it is prime or not. A prime number is a number which is only **divisible by 1 and itself**.

Example 1:

Input:
N = 5
Output:
1
Explanation:
5 has 2 factors 1 and 5 only.

Example 2:

Input:
N = 25
Output:
0
Explanation:
25 has 3 factors 1, 5, 25

Your Task:
You don't need to read input or print anything. Your task is to complete the function **isPrime()** which takes an integer **N** as input parameters and returns an integer, 1 if N is a prime number or 0 otherwise.

Expected Time Complexity: O(sqrt(N))
Expected Space Complexity: O(1)

Constraints:
$1 <= N <= 10^9$

```cpp
/*
    https://www.geeksforgeeks.org/problems/prime-number2314/1
*/

/* Driver code
#include <bits/stdc++.h>
using namespace std;
*/
int isPrime(int N) {
    if (N <= 1) {
        return 0;
    }
    /* Every number has two multiples, one smaller than
    its square root, one greater than its square root.
    If its's not prime it will have a factor less than sqrt(n).
    So, rather than iterating till n,
    you can check until sqrt(n)
    */
    for (int i = 2; i <= sqrt(N); i++) {
        if (N % i == 0) {
            return 0;
        }
    }
    return 1;
}
 /* Driver code
int main() {
    // your code goes here
    return 0;
}
*/
```