FizzBuzz

Given a number *n*, for each integer *i* in the range from *1* to *n* inclusive, print one value per line as follows:

* If *i* is a multiple of both *3* and *5*, print *FizzBuzz*.
* If *i* is a multiple of *3* (but not *5*), print *Fizz*.
* If *i* is a multiple of *5* (but not *3*), print *Buzz*.
* If *i* is not a multiple of *3* or *5*, print the value of *i*.

**Function Description**

Complete the function *fizzBuzz* in the editor below.

fizzBuzz has the following parameter(s):

*int n:*  upper limit of values to test (inclusive)

Returns:    NONE

Prints:

    The function must print the appropriate response for each value *i* in the set *{1, 2, ... n}* in ascending order, each on a separate line.

**Constraints**

* *0 < n < 2 × 105*

**Input from stdin will be processed as follows and passed to the function.**

The single integer *n*, the limit of the range to test: [1, 2, ...n].

**Sample Case 0**

**Sample Input**

STDIN    Function

-----    --------

15    → n = 15

**Sample Output**

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

**Explanation**

The numbers *3*, *6*, *9*, and *12* are multiples of *3* (but not *5*), so print *Fizz* on those lines.

The numbers *5* and *10* are multiples of *5* (but not *3*), so print *Buzz* on those lines.

The number *15* is a multiple of both *3* and *5*, so print *FizzBuzz* on that line.

None of the other values is a multiple of either *3* or *5*, so print the value of *i* on those lines.

**Hint 1**

What math operator is used to check if a number is a divisor of another? Answer: modulo division operator

What is the test for a divisor using the modulo operator? Answer: the result (the remainder of integer division) is zero

**Hint 2**

How do you make your loop start at 1 and end at n, inclusive?

In C: for (int i = 1; i <= n; i++). In Python: for i in range(1, n+1)

Solution

****Skills:**** Loops and counters, I/O

****Brute Force Approach:**** Only a brute force solution is necessary or possible.

Loop through the values from 1 to n, testing if 3, 5 or both are divisors. A common approach has 4 conditions and 4 division operations:

if divisible by 3 and divisible by 5

    print FizzBuzz

if divisible by 3

    print Fizz

if divisible by 5

    print Buzz

else

    print number

****Optimal Solution:****

One minor efficiency is available. By concatenating the answer string in the correct order, the number of conditions can be reduced to 3. Also, divisions are reduced to 2.

import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.function.\*;

import java.util.regex.\*;

import java.util.stream.\*;

import static java.util.stream.Collectors.joining;

import static java.util.stream.Collectors.toList;

class Result {

/\*

\* Complete the 'fizzBuzz' function below.

\*

\* The function accepts INTEGER n as parameter.

\*/

public static void fizzBuzz(int n) {

// Write your code here

}

}

public class Solution {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

int n = Integer.parseInt(bufferedReader.readLine().trim());

Result.fizzBuzz(n);

bufferedReader.close();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*