# Problem 1116: Print Zero Even Odd

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 0.00%
**Paid Only:** No

## Problem Description

You have a function

printNumber

that can be called with an integer parameter and prints it to the console.

For example, calling

printNumber(7)

prints

7

to the console.

You are given an instance of the class

ZeroEvenOdd

that has three functions:

zero

,

even

, and

odd

. The same instance of

ZeroEvenOdd

will be passed to three different threads:

Thread A:

calls

zero()

that should only output

0

's.

Thread B:

calls

even()

that should only output even numbers.

Thread C:

calls

odd()

that should only output odd numbers.

Modify the given class to output the series

"010203040506..."

where the length of the series must be

2n

.

Implement the

ZeroEvenOdd

class:

ZeroEvenOdd(int n)

Initializes the object with the number

n

that represents the numbers that should be printed.

void zero(printNumber)

Calls

printNumber

to output one zero.

void even(printNumber)

Calls

printNumber

to output one even number.

void odd(printNumber)

Calls

printNumber

to output one odd number.

Example 1:

Input:

n = 2

Output:

"0102"

Explanation:

There are three threads being fired asynchronously. One of them calls zero(), the other calls even(), and the last one calls odd(). "0102" is the correct output.

Example 2:

Input:

n = 5

Output:

"0102030405"

Constraints:

1 <= n <= 1000

## Code Snippets

### C++:

```
class ZeroEvenOdd {
private:
int n;

public:
ZeroEvenOdd(int n) {
this->n = n;
}

// printNumber(x) outputs "x", where x is an integer.
void zero(function<void(int)> printNumber) {

}

void even(function<void(int)> printNumber) {

}

void odd(function<void(int)> printNumber) {

}
};
```

### Java:

```
class ZeroEvenOdd {
private int n;

public ZeroEvenOdd(int n) {
this.n = n;
}

// printNumber.accept(x) outputs "x", where x is an integer.
public void zero(IntConsumer printNumber) throws InterruptedException {

}
public void even(IntConsumer printNumber) throws InterruptedException {

}

public void odd(IntConsumer printNumber) throws InterruptedException {

}
}
```

**Python3:**

```
class ZeroEvenOdd:
def __init__(self, n):
self.n = n



# printNumber(x) outputs "x", where x is an integer.
def zero(self, printNumber: 'Callable[[int], None]') -> None:




def even(self, printNumber: 'Callable[[int], None]') -> None:




def odd(self, printNumber: 'Callable[[int], None]') -> None:
```

**Python:**

```
class ZeroEvenOdd(object):
def __init__(self, n):
self.n = n



# printNumber(x) outputs "x", where x is an integer.
def zero(self, printNumber):
"""
:type printNumber: method
:rtype: void
"""
```

```
def even(self, printNumber):
"""
:type printNumber: method
:rtype: void
"""



def odd(self, printNumber):
"""
:type printNumber: method
:rtype: void
"""
```

### C#:

```
public class ZeroEvenOdd {
private int n;

public ZeroEvenOdd(int n) {
this.n = n;
}

// printNumber(x) outputs "x", where x is an integer.
public void Zero(Action<int> printNumber) {

}

public void Even(Action<int> printNumber) {

}

public void Odd(Action<int> printNumber) {

}
}
```

### C:

```
typedef struct {
int n;
} ZeroEvenOdd;

ZeroEvenOdd* zeroEvenOddCreate(int n) {
ZeroEvenOdd* obj = (ZeroEvenOdd*) malloc(sizeof(ZeroEvenOdd));
obj->n = n;
return obj;
}

// You may call global function `void printNumber(int x)`
// to output "x", where x is an integer.

void zero(ZeroEvenOdd* obj) {

}

void even(ZeroEvenOdd* obj) {

}

void odd(ZeroEvenOdd* obj) {

}

void zeroEvenOddFree(ZeroEvenOdd* obj) {

}
```

## Solutions

### C++ Solution:

```
/*
* Problem: Print Zero Even Odd
* Difficulty: Medium
* Tags: general
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
```

```
*/

class ZeroEvenOdd {
private:
int n;

public:
ZeroEvenOdd(int n) {
this->n = n;
}

// printNumber(x) outputs "x", where x is an integer.
void zero(function<void(int)> printNumber) {

}

void even(function<void(int)> printNumber) {

}

void odd(function<void(int)> printNumber) {

}
};
```

### Java Solution:

```
/**
* Problem: Print Zero Even Odd
* Difficulty: Medium
* Tags: general
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

class ZeroEvenOdd {
private int n;

public ZeroEvenOdd(int n) {
```

```
this.n = n;
}

// printNumber.accept(x) outputs "x", where x is an integer.
public void zero(IntConsumer printNumber) throws InterruptedException {

}

public void even(IntConsumer printNumber) throws InterruptedException {

}

public void odd(IntConsumer printNumber) throws InterruptedException {

}
}
```

### Python3 Solution:

```
"""
Problem: Print Zero Even Odd
Difficulty: Medium
Tags: general

Approach: Optimized algorithm based on problem constraints
Time Complexity: O(n) to O(n^2) depending on approach
Space Complexity: O(1) to O(n) depending on approach
"""

class ZeroEvenOdd:
def __init__(self, n):
self.n = n



# printNumber(x) outputs "x", where x is an integer.
def zero(self, printNumber: 'Callable[[int], None]') -> None:
# TODO: Implement optimized solution
pass
```

### Python Solution:

```
class ZeroEvenOdd(object):
def __init__(self, n):
self.n = n


# printNumber(x) outputs "x", where x is an integer.
def zero(self, printNumber):
"""
:type printNumber: method
:rtype: void
"""



def even(self, printNumber):
"""
:type printNumber: method
:rtype: void
"""



def odd(self, printNumber):
"""
:type printNumber: method
:rtype: void
"""
```

### C# Solution:

```
/*
* Problem: Print Zero Even Odd
* Difficulty: Medium
* Tags: general
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

public class ZeroEvenOdd {
```

```
private int n;

public ZeroEvenOdd(int n) {
this.n = n;
}

// printNumber(x) outputs "x", where x is an integer.
public void Zero(Action<int> printNumber) {

}

public void Even(Action<int> printNumber) {

}

public void Odd(Action<int> printNumber) {

}
}
```

### C Solution:

```
/*
* Problem: Print Zero Even Odd
* Difficulty: Medium
* Tags: general
*
* Approach: Optimized algorithm based on problem constraints
* Time Complexity: O(n) to O(n^2) depending on approach
* Space Complexity: O(1) to O(n) depending on approach
*/

typedef struct {
int n;
} ZeroEvenOdd;

ZeroEvenOdd* zeroEvenOddCreate(int n) {
ZeroEvenOdd* obj = (ZeroEvenOdd*) malloc(sizeof(ZeroEvenOdd));
obj->n = n;
return obj;
}
```

```
// You may call global function `void printNumber(int x)`
// to output "x", where x is an integer.

void zero(ZeroEvenOdd* obj) {

}

void even(ZeroEvenOdd* obj) {

}

void odd(ZeroEvenOdd* obj) {

}

void zeroEvenOddFree(ZeroEvenOdd* obj) {

}
```