Generate 0 and 1 with 25% and 75% probability

Difficulty Level : Medium • Last Updated : 06 Nov, 2021

Given a function rand50() that returns 0 or 1 with equal probability, write a function that returns 1 with 75% probability and 0 with 25% probability using rand50() only. Minimize the number of calls to rand50() method. Also, use of any other library function and floating point arithmetic are not allowed.

Recommended: Please try your approach on ***{IDE}*** first, before moving on to the solution.

The idea is to use **Bitwise OR**. A bitwise OR takes two bits and returns 0 if both bits are 0, while otherwise the result is 1. So it has 75% probability that it will return 1.
Below is the implementation of above idea :

C++

```
// Program to print 1 with 75% probability and 0
// with 25% probability
#include <iostream>
using namespace std;

// Random Function to that returns 0 or 1 with
// equal probability
```

```
// probability and 0 with 25% probability using
// Bitwise OR
bool rand75()
{
    return rand50() | rand50();
}

// Driver code to test above functions
int main()
{
    // Initialize random number generator
    srand(time(NULL));

    for(int i = 0; i < 50; i++)
        cout << rand75();

    return 0;
}
```

Java

```
// Java program to print 1 with 75% probability and 0
// with 25% probability
class GFG
{

    // Random Function to that returns 0 or 1 with
    // equal probability
    static int rand50()
    {
        // rand() function will generate odd or even
        // number with equal probability. If rand()
        // generates odd number, the function will
        // return 1 else it will return 0.
        return (int) (10 * Math.random()) & 1;
    }

    // Random Function to that returns 1 with 75%
```

```
    {
        // Initialize random number generator
        //srand(time(null));

        for (int i = 0; i < 50; i++)
        {
            System.out.print(rand75());
        }

    }

}

// This code is contributed by 29AjayKumar
```

PHP

```
<?php
// Program to print 1 with 75% probability
// and 0 with 25% probability

// Random Function to that returns 0 or
// 1 with equal probability
function rand50()
{

    // rand() function will generate
    // odd or even number with equal
    // probability. If rand() generates
    // odd number, the function will
    // return 1 else it will return 0.
    return rand() & 1;
}

// Random Function to that returns
// 1 with 75% probability and 0
// with 25% probability using
// Bitwise OR
```

```
    for($i = 0; $i < 50; $i++)
        echo rand75();

// This code is contributed m_kit
?>
```

Javascript

```
<script>
// Program to print 1 with 75% probability and 0
// with 25% probability


// Random Function to that returns 0 or 1 with
// equal probability
function rand50() {
    // rand() function will generate odd or even
    // number with equal probability. If rand()
    // generates odd number, the function will
    // return 1 else it will return 0.
    return Math.floor(Math.random() * 10) & 1;
}

// Random Function to that returns 1 with 75%
// probability and 0 with 25% probability using
// Bitwise OR
function rand75() {
    return rand50() | rand50();
}

// Driver code to test above functions

// Initialize random number generator


for (let i = 0; i < 50; i++)
    document.write(rand75());
```

```
11101111110010010110011111111101111110111100011000
```

Time Complexity: *O(1)*

Auxiliary Space: *O(1)*

On similar lines, we can also use **Bitwise AND**. Since it returns 0 with 75% probability, we have to invert the result.

```
// Random Function to that returns 1 with 75%
// probability and 0 with 25% probability using
// Bitwise AND
bool rand75()
{
    return !(rand50() & rand50());
}
```

We can replace Bitwise OR and Bitwise AND operator by **OR and AND operators** as well -

```
// Random Function to that returns 1 with 75%
// probability and 0 with 25% probability using
// OR or AND operator
int rand75()
{
    return !(rand50() && rand50());
    // return rand50() || rand50()
```

```
// Program to print 1 with 75% probability and 0
// with 25% probability
#include <iostream>
using namespace std;

// Random Function to that returns 0 or 1 with
// equal probability
int rand50()
{
    // rand() function will generate odd or even
    // number with equal probability. If rand()
    // generates odd number, the function will
    // return 1 else it will return 0.
    return rand() & 1;
}

// Random Function to that returns 1 with 75%
// probability and 0 with 25% probability using
// left shift and Bitwise XOR
int rand75()
{
    // x is one of {0, 1}
    int x = rand50();

    x = x << 1;

    // x is now one of {00, 10}

    x = x ^ rand50();

    // x is now one of {00, 01, 10, 11}

    return (x > 0) ? 1 : 0;
}

// Driver code to test above functions
int main()
{
    // Initialize random number generator
```

Java

```
// Java program to print 1 with 75% probability and 0
// with 25% probability
class GFG
{

// Random Function to that returns 0 or 1 with
// equal probability
static int rand50()
{
    // rand() function will generate odd or even
    // number with equal probability. If rand()
    // generates odd number, the function will
    // return 1 else it will return 0.
    return (int) (10 * Math.random()) & 1;
}

// Random Function to that returns 1 with 75%
// probability and 0 with 25% probability using
// left shift and Bitwise XOR
static int rand75()
{
    // x is one of {0, 1}
    int x = rand50();

    x = x << 1;

    // x is now one of {00, 10}

    x = x ^ rand50();

    // x is now one of {00, 01, 10, 11}

    return (x > 0) ? 1 : 0;
}

// Driver code
public static void main(String[] args)
```

PHP

```php
<?php
// Program to print 1 with
// 75% probability and 0
// with 25% probability

// Random Function to that
// returns 0 or 1 with
// equal probability
function rand50()
{
    // rand() function will
    // generate odd or even
    // number with equal
    // probability. If rand()
    // generates odd number,
    // the function will return
    // 1 else it will return 0.
    return rand() & 1;
}

// Random Function to that
// returns 1 with 75%
// probability and 0 with
// 25% probability using
// left shift and Bitwise XOR
function rand75()
{
    // x is one of {0, 1}
    $x = rand50();

    $x = $x << 1;

    // x is now one
    // of {00, 10}

    $x = $x ^ rand50();
```

```
// number generator
srand(time(NULL));

for ($i = 0; $i < 50; $i++)
    echo rand75();

// This code is contributed
// by ajit
?>
```

Javascript

```
<script>
// Javascript program to print 1 with 75% probability and 0
// with 25% probability

    // Random Function to that returns 0 or 1 with
    // equal probability
function rand50()
{

        // rand() function will generate odd or even
        // number with equal probability. If rand()
        // generates odd number, the function will
        // return 1 else it will return 0.
        return  Math.floor((10 * Math.random())) & 1;
}

    // Random Function to that returns 1 with 75%
    // probability and 0 with 25% probability using
    // Bitwise OR
function rand75()
{
    // x is one of {0, 1}
    let x = rand50();

    x = x << 1;
```

```
// Driver code


for (let i = 0; i < 50; i++)
{
    document.write(rand75());
}

// This code is contributed by rag2127
</script>
```

Output:

```
01101110111011000111111111110001111011101110110110
```

Time Complexity: O(1)

Auxiliary Space: O(1)

Please note above solutions will produce **different results** every time we run them.
This article is contributed by **Aditya Goel**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.
Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Previous

Next

RECOMMENDED ARTICLES

Page : 1 2 3

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Medium

Easy Normal Medium Hard Expert

Improved By : jit_t, 29AjayKumar, Akanksha_Rai, gfgking, rag2127, subhammahato348

Article Tags : Bitwise-XOR, Bit Magic, Randomized

Practice Tags : Bit Magic

Improve Article Report Issue