

# Techniques for Contest

## Even-Odd

bitwise AND (&) operator দিয়ে efficient ভাবে জোড়-বিজোড় বের করা যায়। নিচে এরকম একটি প্রোগ্রাম দেখানো হল:

```
#include<cstdio>
#include<iostream>

using namespace std;

int main()
{
    int x[10];

    for(int i=0; i<10; i++)
        x[i] = i;

    for(int i=0; i<10; i++)
    {
        if(x[i]&1)
            cout << i << " is ODD." << endl;
        else
            cout << i << " is EVEN." << endl;
    }

    return 0;
}

/// Why it works?

/// let a value: 5
/// Here, 5 = 0000101
/// So, 5 & 1 =>
/// 0000101 = 5
/// 0000001 = 1
/// -----
/// 0000001 = 1 , that represent true.

/// On the other hand,
/// let a value: 8
/// So, 8 = 00001000
/// So, 8 & 1 =>
/// 00001000 = 8
/// 00000001 = 1
/// -----
/// 00000000 = 0 , that represent false.

/// This is because when converting to base 10, the rightmost bit represents 2^0 or 1. When the
rightmost bit is 1, we know that our number is odd since we're adding 1 to a bunch of powers of two
which will always be even. When the rightmost bit is 0, we know our number will be even, since it
simply consists of adding up a bunch of even numbers.
```

প্রতিটি সংখ্যার শেষ বিটটি (Least Significant Bit) জোড় হলে ০ এবং বিজোড় হলে ১ হয়। তাই উপরের পদ্ধতিটি দিয়ে যে কোনো সংখ্যা জোড়-বিজোড় পরীক্ষা করা যায় এবং bitwise অপারেটর অপারেশন গুলো অন্য অপারেটরের তুলনায় অনেক efficient.  
(2015/11/06 06:06)