Fizz Buzz

写一个程序，输出从 1 到 *n* 数字的字符串表示。

1. 如果 *n* 是3的倍数，输出“Fizz”；

2. 如果 *n* 是5的倍数，输出“Buzz”；

3.如果 *n* 同时是3和5的倍数，输出 “FizzBuzz”。

示例：

```
n = 15,
```

返回：

```
[
    "1",
    "2",
    "Fizz",
    "4",
    "Buzz",
    "Fizz",
    "7",
    "8",
    "Fizz",
    "Buzz",
    "11",
    "Fizz",
    "13",
    "14",
    "FizzBuzz"
]
```

我的解法；

```
public class Solution {
    public IList<string> FizzBuzz(int n) {
    List<string> result = new List<string>();
      for(int i=1;i<=n;i++)
      {
          if(i%3==0){
              if(i%5==0)
              {
                  result.Add("FizzBuzz");
              }
              else
              {
                  result.Add("Fizz");
              }
          }
          else if(i%5==0)
```

```
            {
                result.Add("Buzz");
            }
            else{
                 result.Add(i.ToString());
               }
        }
          return  result;
    }
}
```

计数质数

统计所有小于非负整数 *n* 的质数的数量。

示例:

输入：10

输出：4

解释：小于 10 的质数一共有 4 个，它们是 2，3，5，7 。

```
public class Solution {
    public int CountPrimes(int n) {
        int count = 0;
        //初始全都false
        bool []flag = new bool[n];
        //从2开始循环
        for (int i = 2; i < n ; i++)
            //为false则将其定义为可能为质数
            if (flag[i] == false){
                count ++;
                for (int j = 1; j * i < n ; j++)
                    flag[j * i] = true;
            }
        return count;
    }
}
```

3的幂

给定一个整数，写一个函数来判断它是否是 3 的幂次方。

示例 1:

输入： `27`
输出： `true`
示例 2:
输入： `0`
输出： `false`
示例 3:
输入： `9`
输出： `true`
示例 4:
输入： `45`
输出： `false`
进阶：
你能不使用循环或者递归来完成本题吗?

```
public class Solution {
    public bool IsPowerOfThree(int n) {
        if(n<=0) return false;
        while(n>1)
        {
        if(n%3!=0)
            {
            return false;
            }
            n=n/3;
        }
        return true;
    }
}
```

位1的个数

编写一个函数，输入是一个无符号整数，返回其二进制表达式中数字位数为 ‘1’ 的个数（也被称为[汉明重量](https://baike.baidu.com/item/%E6%B1%89%E6%98%8E%E9%87%8D%E9%87%8F)）。
示例：
输入： `11`
输出： `3`

解释：整数 `11` 的二进制表示为 `00000000000000000000000000001011`

示例 2:

输入：`128`
输出：`1`
解释：整数 `128` 的二进制表示为 `00000000000000000000000010000000`

我的解法:

```
public class Solution {
    public int HammingWeight(uint n) {
       var flag = Convert.ToString(n,2);
       int Count=0;
       for(int i=0;i<flag.Length;i++)
       {
           if(flag[i]=='1')
           {
               Count=Count+1;
           }
       }
        return Count;
    }
}
```