

E So 2——Officially

Description

在学了计组之后，Thor 终于认识到原来所有的整数在计算机里面都是用 2 进制储存的！好了，你来把一个数 x 拆分成 2 的幂的和的形式吧，要求数尽量少，且从小到大输出。

Input

多组数据。

每组数据仅有 1 行 1 个整数 x ($\text{MAX_INT} \geq x \geq 1$)

Output

把 x 拆分成 2 的幂的形式，数尽量少，从小到大输出。

Sample Input

```
3
4
5
6
```

Sample Output

```
1 2
4
1 4
2 4
```

Hint

解题分析

本题想不通则已，一旦想通，代码长度巨短无比。仔细考虑二进制与十进制的转换方式，比如现有一二进制数：1111，我们转换的时候就是 $1*2^0+1*2^1+1*2^2+1*2^3$ ，即 $1+2+4+8$ 从而得到了 15。

归纳整理一下：

一个二进制数转化成十进制，即十进制拆分为 2 的幂次的转换公式如下：

$$DEC(N) = \sum_{i=0}^n 2^i * BIN(N)[i]$$

其中 DEC(N) 表示 N 在十进制下的表示，BIN(N) 表示 N 在二进制下的表示，n 为二进制表示的位数，i 为二进制表示的第 i 位，从右往左依次为第 0 位，第 1 位，...，第 n-1 位。

于是我们的算法就出来了——反向，将十进制数每次除以二，同时使用一个变量 x 记录二进制第 i 位对应 2 的求幂结果，若除以二的余数为 1，则输出，否则继续计算。具体参见参考代码。

（小小的吐个槽：说好的 HINT 呢？！）

参考代码：

标准版

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    while(cin>>n) {
        int x=1;
        while(n) {
            if(n%2) cout<<x<<' ';
            x*=2;    n/=2;
        }
        cout<<endl;
    }
}
```

位运算版

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    while(cin>>n) {
        int x=1;
        while(n) {
            if(n&1) cout<<x<<' ';
            x*=2;    n>>=1;
        }
        cout<<endl;
    }
}
```