# 题目

二进制数转字符串。给定一个介于0和1之间的实数（如0.72），类型为double，打印它的二进制表达式。如果该数字无法精确地用32位以内的二进制表示，则打印“ERROR”。

示例1:

输入：0.625

输出："0.101"

示例2:

输入：0.1

输出："ERROR"

提示：0.1无法被二进制准确表示

提示：

32位包括输出中的 "0." 这两位。

题目保证输入用例的小数位数最多只有 6 位

# 分析

## 方法一：模拟

这个问题可以通过乘以2并检查整数部分的方法来转换为二进制表示。具体步骤如下：

1、将小数部分乘以2，如果乘积大于等于1，则整数部分为1，否则为0。

2、将整数部分加到结果字符串中，并将乘积减去整数部分。

3、重复上述步骤，直到小数部分为0或者结果字符串长度达到32。

在实现中，需要注意小数部分可能出现无限循环的情况，因此需要限制循环次数。以下是实现代码：

class Solution {

public:

string printBin(double num) {

string result = "0.";

int count = 0;

while (num > 0) {

if (count >= 32) {

return "ERROR";

}

num \*= 2;

if (num >= 1) {

result += '1';

num -= 1;

} else {

result += '0';

}

count++;

}

return result;

}

};

或：

class Solution {

public:

string printBin(double num) {

string res = "0.";

while (res.size() <= 32 && num != 0) {

num \*= 2;

int digit = num;

res.push\_back(digit + '0');

num -= digit;

}

return res.size() <= 32 ? res : "ERROR";

}

};