

## 問題一

### 問題描述

小蘇最近在學三種邏輯運算子 AND、OR 和 XOR。這三種運算子都是二元運算子，也就是說在運算時需要兩個運算元，例如  $a \text{ AND } b$ 。對於整數  $a$  與  $b$ ，以下三個二元運算子的運算結果定義如下列三個表格：

a AND b		
	b 為 0	b 不為 0
a 為 0	0	0
a 不為 0	0	1

a OR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	1

a XOR b		
	b 為 0	b 不為 0
a 為 0	0	1
a 不為 0	1	0

舉例來說：

- (1)  $0 \text{ AND } 0$  的結果為 0， $0 \text{ OR } 0$  以及  $0 \text{ XOR } 0$  的結果也為 0。
- (2)  $0 \text{ AND } 3$  的結果為 0， $0 \text{ OR } 3$  以及  $0 \text{ XOR } 3$  的結果則為 1。
- (3)  $4 \text{ AND } 9$  的結果為 1， $4 \text{ OR } 9$  的結果也為 1，但  $4 \text{ XOR } 9$  的結果為 0。

請撰寫一個程式，讀入  $a$ 、 $b$  以及邏輯運算的結果，輸出可能的邏輯運算為何。

### 輸入格式

輸入只有一行，共三個整數值，整數間以一個空白隔開。第一個整數代表  $a$ ，第二個整數代表  $b$ ，這兩數均為非負的整數。第三個整數代表邏輯運算的結果，只會是 0 或 1。

### 輸出格式

輸出可能得到指定結果的運算，若有多個，輸出順序為 AND、OR、XOR，每個可能的運算單獨輸出一行，每行結尾皆有換行。若不可能得到指定結果，輸出 IMPOSSIBLE。（注意輸出時所有英文字母均為大寫字母。）

範例一：輸入

0 0 0

範例一：正確輸出

AND

OR

XOR

範例三：輸入

3 0 1

範例三：正確輸出

OR

XOR

範例二：輸入

1 1 1

範例二：正確輸出

AND

OR

範例四：輸入

0 0 1

範例四：正確輸出

IMPOSSIBLE

### 評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制(time limit)均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 80 分， $a$  和  $b$  的值只會是 0 或 1。

第 2 子題組 20 分， $0 \leq a, b < 10,000$ 。

## 問題二

### 問題描述

一次考試中，於所有及格學生中獲取最低分數者最為幸運，反之，於所有不及格同學中，獲取最高分數者，可以說是最為不幸，而此二種分數，可以視為成績指標。

請你設計一支程式，讀入全班成績(人數不固定)，請對所有分數進行排序，並分別找出不及格中最高分數，以及及格中最低分數。

當找不到最低及格分數，表示對於本次考試而言，這是一個不幸之班級，此時請你印出：「worst case」；反之，當找不到最高不及格分數時，請你印出「best case」。

註：假設及格分數為 60，每筆測資皆為 0~100 間整數，且筆數未定。

### 輸入格式

第一行輸入學生人數，第二行為各學生分數(0~100 間)，分數與分數之間以一個空白間格。每一筆測資的學生人數為 1~20 的整數。

### 輸出格式

每筆測資輸出三行。

第一行由小而大印出所有成績，兩數字之間以一個空白間格，最後一個數字後無空白；

第二行印出最高不及格分數，如果全數及格時，於此行印出 best case；

第三行印出最低及格分數，當全數不及格時，於此行印出 worst case。

### 範例一：輸入

```
10
0 11 22 33 55 66 77 99 88 44
```

### 範例一：正確輸出

```
0 11 22 33 44 55 66 77 88 99
55
66
```

(說明) 不及格分數最高為 55，及格分數最低為 66。

### 範例二：輸入

```
1
13
```

### 範例二：正確輸出

```
13
13
worst case
```

(說明) 由於找不到最低及格分，因此第三行須印出「worst case」。

範例三：輸入

```
2
73 65
```

範例三：正確輸出

```
65 73
best case
65
```

（說明）由於找不到不及格分，因此第二行須印出「best case」。

### 問題3

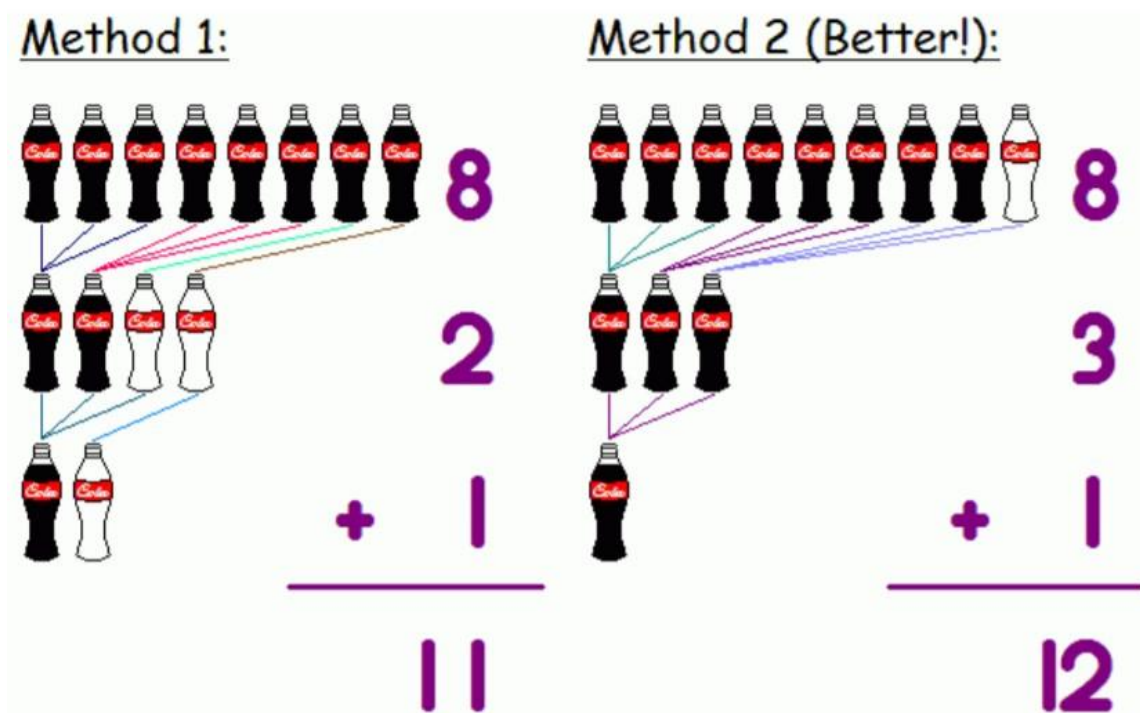
You see the following special offer by the convenience store:

*"A bottle of Choco Cola for every 3 empty bottles returned"*

Now you decide to buy some (say  $N$ ) bottles of cola from the store. You would like to know how you can get the most cola from them.

The figure below shows the case where  $N = 8$ . *Method 1* is the standard way: after finishing your 8 bottles of cola, you have 8 empty bottles. Take 6 of them and you get 2 new bottles of cola. Now after drinking them you have 4 empty bottles, so you take 3 of them to get yet another new cola. Finally, you have only 2 bottles in hand, so you cannot get new cola any more. Hence, you have enjoyed  $8 + 2 + 1 = 11$  bottles of cola.

You can actually do better! In *Method 2*, you first borrow an empty bottle from your friend (! Or the storekeeper??), then you can enjoy  $8 + 3 + 1 = 12$  bottles of cola! Of course, you will have to return your remaining empty bottle back to your friend.



#### Input

Input consists of several lines, each containing an integer  $N$  ( $1 \leq N \leq 200$ ).

#### Output

For each case, your program should output the maximum number of bottles of cola you can enjoy. You may borrow empty bottles from others, but if you do that, make sure that you have enough bottles afterwards to return to them.

**Note:** Drinking too much cola is bad for your health, so... don't try this at home!! :-)

#### Sample Input

8

#### Sample Output

12