

Problem A. 一个简单的位运算

题目描述

有一只蒟蒻在进行位运算，对于两个二进制数 x, y ，它想要知道 $x \oplus y, x \mid y, x \& y$ 分别是多少。

注意：本题以及下一题中的 \oplus 均表示位运算中的异或运算。

输入格式

三行，第一行一个整数 n ，表示二进制数的位数。

第二行和第三行每行 n 个用空格隔开的数（0 或 1），表示两个二进制数 x, y 的每一位（从高位到低位）。

输出格式

三行，每行一个 n 位二进制数，分别表示 $x \oplus y, x \mid y, x \& y$ 。

样例输入

```
3
0 0 1
1 1 1
```

样例输出

```
110
111
001
```

数据范围

$n \leq 100$

Hint

还记得数组怎么用吗~

作为友好的第一题给一个输入 n 个数到数组的小提示：

```
int i, a[105];
for(i=0; i<n; i++)
    scanf("%d", &a[i]);
```

接下来相信你一定还记得位运算的规则是什么~

Author YUKILSY

Problem B. 又是一个简单的位运算

题目描述

有一只蒟蒻又在进行位运算，对于 a_1, a_2, \dots, a_n ， $a_i \in \{0, 1\}$ ，它想要计算如下内容：

$$\begin{aligned} & a_1 \oplus a_2 \oplus a_3 \cdots \oplus a_n \\ & a_1 \mid a_2 \mid \cdots \mid a_n \\ & a_1 \& a_2 \& \cdots \& a_n \end{aligned}$$

但是由于 n 实在是太大了，以及这只蒟蒻实在是太弱了，因此ta只知道 a_1, a_2, \dots, a_n 中有多少个1。

输入格式

本题存在多组数据，对于每组数据：

一行，两个用空格分隔的整数 n, x ，表示 a_1, a_2, \dots, a_n 中有 x 个1。（那 a_1, a_2, \dots, a_n 中自然就有 $n - x$ 个0）

输出格式

对于每组数据：

一行，三个空格分隔的整数，分别表示 $a_1 \oplus a_2 \oplus a_3 \cdots \oplus a_n$ 、 $a_1 \mid a_2 \mid \cdots \mid a_n$ 、 $a_1 \& a_2 \& \cdots \& a_n$ 。

样例输入

```
5 3
6 4
```

样例输出

```
1 1 0
0 1 0
```

样例解释

$$1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$1 \mid 1 \mid 1 \mid 0 \mid 0 = 1$$

$$1 \& 1 \& 1 \& 0 \& 0 = 0$$

数据范围

对于100%的数据 $1 \leq n \leq 10^4$ ，数据组数 $\leq 10^3$ 。

Hint

众所周知位运算满足交换律~

但这道题或许有着非常简单，代码非常短的做法呢~

以及还记得多组数据怎么读入吗~

```
while(scanf("%d%d",&n,&x)!=EOF){
    // do something
}
```

Author YUKILSY

Problem C. MYX的函数

题目描述

MYXgiegie获得了这样一个字符串函数：

$$f(x, y) = z$$

具体而言， x, y 是两个长度均为 n 的只由小写字母组成的输入字符串， z 是长度为 n 的输出字符串。这个函数对字符串进行了如下操作：

$$z[i] = \min(x[i], y[i]), i \in [0, n)$$

现在，请你编程实现这个函数吧！

输入格式

第一行为字符串长度 n ，($1 \leq n \leq 30$)。

第二行为输入的 x, y ，具体而言，第 $2i$ 个字符是 x 的第 i 个字符，第 $2i+1$ 个字符是 y 的第 i 个字符， $i \in [0, n-1]$ 。

输出格式

输出字符串 z ，每个字符间有一个空格，字符串末尾没有空格。

样例输入

```
3
mmyxxy
```

样例输出

```
m x x
```

样例说明

无

Hint

注意：输出字符串末尾没有空格哦

请使用scanf("%d\n",&n);的方式输入第一行的整数。（可以思考一下是为什么呢~）

AUTHOR : ghy

Problem D. 最大数字频数统计

题目描述

现在有 n 个数字，请你找出其中的最大数字，并统计该数字出现的次数。

输入格式

输入共两行。

第一行为一个正整数 n ($1 \leq n \leq 100$)，表示数字的数量。

第二行为 n 个 `int` 范围内的数字 a_i ($1 \leq i \leq n$)，用空格分隔。

输出格式

输出共两行。

第一行为最大数字。

第二行为最大数字出现的次数。

样例输入1

```
6
1 2 3 4 5 6
```

样例输出1

```
6
1
```

样例输入2

```
6
3 3 -1 2 3 3
```

样例输出2

```
3
4
```

Hint

`int` 范围最大值为2147483647，最小值为-2147483648。

Problem E. Hamming Distance

题目描述

在信息论中，Hamming Distance 表示两个等长字符串在对应位置上不同字符的数目。

现在，给出两个整数a,b，请你计算这两个整数在二进制表示中的 Hamming Distance。

输入格式

一行，两个非负数a,b。 ($0 \leq a, b \leq 2^{64} - 1$)

输出格式

一个数字，为这两个整数在64位二进制表示下的 Hamming Distance。

样例输入1

```
1 1
```

样例输出1

```
0
```

样例输入2

```
1 0
```

样例输出2

```
1
```

Hint

本题需要的数据范围为 $2^{64} - 1$ ，需要用到unsigned long long类型，该类型输入输出的占位符为“%llu”。

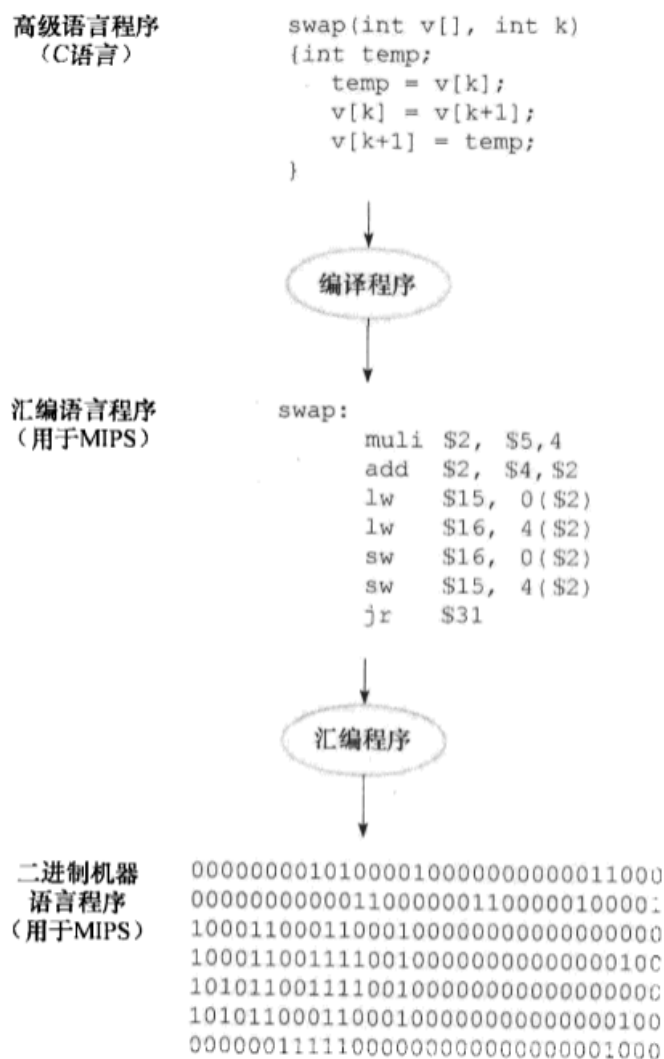
AUTHOR : 爱出水题的ghy

Problem F. 模拟汇编

题目背景

跳过这个长长的背景也不会影响做题，但本题的背景是个很有意思的科普呀~

众所周知，计算机的底层是读不懂 C 语言这种高级语言的，我们写的 C 语言程序，首先被编译成汇编语言程序，再被汇编成计算机底层可以看懂并执行的二进制机器语言程序。



本题进行最简单的模拟汇编。

汇编语言程序只能操作放在寄存器里的数，不能直接操作储存在磁盘里的数。

寄存器是 CPU 内部用来存放数据的一些小型存储区域，用来暂时存放参与运算的数据和运算结果。典型现代计算机中有 32 个寄存器，分别为 0 号寄存器~31 号寄存器，分别用 \$0 到 \$31 表示。

如果我们想把 8 号寄存器里的数和 9 号寄存器里的数字加起来放到 10 号寄存器里，即高级语言的 `$10=$9+$8`，MIPS 汇编语言指令如下。

```
add $10,$9,$8
```

同理，如果我们想把 8 号寄存器里的数和 9 号寄存器里的数字相减放到 10 号寄存器里，即高级语言的 `$10=$9-$8`，MIPS 汇编语言指令如下。

```
sub $10,$9,$8
```

一条 MIPS 指令被汇编成二进制机器语言以后，在计算机中占 32 个二进制位。本题只讨论 add 指令。如 add `$10,$9,$8` 对应的 32 位机器语言是(加了 | 方便大家看)

```
000000|01001|01000|01010|00000|100000
```

1-6 位是操作码，add 指令的操作码恒为 000000，
7-11 位是第一个源操作数的寄存器编号（9）的二进制形式，
12-16 位是第二个源操作数的寄存器编号（8）的二进制形式，
17-21 位是存放计算结果的寄存器编号（10）的二进制形式，
22-26 位是位移量，add 指令的位移量恒为 00000，
27-32 位是功能码，add 指令的功能码恒为 100000。

蕾蕾在书上看到一条 32 位机器语言 add 指令，但是她国庆要出去和小秋月团建，便拜托聪明的你帮她把这句话翻译成汇编语言指令（add \$x,\$y,\$z）。

对了，蕾蕾去 ktv 团建之前并不想说太多话，所以她只会给你 8 位 16 进制数来代替这 32 位二进制数，她相信这难不倒你。

题目描述

给出一个八位十六进制数，请先把它转换成 32 位二进制数，将这个 32 位二进制数的**从左往右数（从高位到低位）** 7~11 位转换成十进制数 y ，12~16 位转换成十进制数 z ，17~21 位转换成十进制数 x ，输出一句话 add \$x,\$y,\$z，具体请看样例分析。

样例输入1

```
01285020
```

样例输出1

```
add $10,$9,$8
```

样例输入2

```
012af020
```

样例输出2

```
add $30,$9,$10
```

样例解释

- 十六进制数 01285020 转换成 32 位二进制数是 00000001001010000101000000100000
第 7 — 11 位是 01001，转换成十进制是 9，第 12 — 16 位是 01000，转换成十进制是 8，第 17 — 21 位是 01010，转换成十进制是 10，所以答案是 add \$10,\$9,\$8。
- 十六进制数 012af020 转换成 32 位二进制数是 00000001001010101111000000100000
第 7 — 11 位是 01001，转换成十进制是 9，第 12 — 16 位是 01010，转换成十进制是 10，第 17 — 21 位是 11110，转换成十进制是 30，所以答案是 add \$30,\$9,\$10。

Hint

一开始的输入十六进制数的环节，不必开 `char a[9]`，只用一个 `int` 就行。
`%x` 可以输入十六进制数字的，可以超大降低难度。

```
int sixteen,i,temp;
scanf("%x",&sixteen);
printf("这个数的十进制是%d\n",sixteen);
for (i =1; i <= 8; i++)
{
    temp = sixteen % 16;
    sixteen /= 16;
    printf("该十六进制数的第%d位代表的数字是%d\n",i,temp);
}
```

输入

7abc9f83

输出

这个数的十进制是2059181955
该十六进制数的第1位代表的数字是3
该十六进制数的第2位代表的数字是8
该十六进制数的第3位代表的数字是15
该十六进制数的第4位代表的数字是9
该十六进制数的第5位代表的数字是12
该十六进制数的第6位代表的数字是11
该十六进制数的第7位代表的数字是10
该十六进制数的第8位代表的数字是7

怎么样，是不是和普通的十进制按位拆分完全一致~

提前祝大家国庆假期愉快，欢迎来到软件学院学计算机硬件基础。

进制转换可以思路清晰地解决这道题目，但如果你的位运算掌握得比较熟练的话，或许有更简单的办法~

AUTHOR bxl

Problem G. 士谔2173&&传源2171

题目描述

士谔书院 21 级的编号是 2173，传源书院 21 级的编号是 2171。所以 GXY 规定：如果一个 10 进制的正整数，它的每一位中出现 2, 1, 7 的次数的和如果为 2 或 1 或 7，那么它就是**士谔传源数**。

如：201 中每一位出现 2, 1, 7 的次数和为 2，2003 中每一位出现 2, 1, 7 的次数和为 1，2117721 中每一位出现 2, 1, 7 的次数和为 7，它们都是**士谔传源数**。2021 中出现 2, 1, 7 的次数和为 3，因此它不是**士谔传源数**。

现在请你输出 $[L, R]$ 中所有的**士谔传源数**。

输入格式

第一行输入数据组数 n ($1 \leq n \leq 10^4$)。

接下来 n 行每行输入两个正整数 L, R ($1 \leq L \leq R \leq 20000$)。

输出格式

对于每组数据，输出一行，为之前**未输出过**的 $[L, R]$ 内的**士谔传源数**，以空格分隔。

样例输入1

```
2
1 5
1 10
```

样例输出1

```
1 2
7 10
```

样例输入2

```
3
1 5
1 5
1 10
```

样例输出2

```
1 2

7 10
```

Hint

怎么判断一个数字之前有没有出现过呢~

或许可以用一个数组标记一下，比如当 x 已经出现时，令 $\text{vis}[x]=1$ 。

如果TLE的话可以思考一下怎么能够预防重复判断某个数字是否是士谔传源数。

Author: gxy

Problem H. 负二进制数

题目描述

相信同学们都已经对二进制数（base 2）有了一定的了解，于是阿彩学长决定召唤二进制的姐妹——负二进制。该进制以 -2 为基数，用 0 和 1 两个数码来表示。一个负二进制数可以表示为每位数码乘以一个以该数字所处位置为指数，以 -2 为底数的幂之和的形式。例如：

$$-15 = (110001)_{-2} = 1 \times (-2)^5 + 1 \times (-2)^4 + 0 \times (-2)^3 + 0 \times (-2)^2 + 0 \times (-2)^1 + 1 \times (-2)^0$$

现给定一个十进制数 n ，请你将其转换为负二进制数（base -2）输出。

输入格式

第一行为 t ($t \leq 100$)，表示数据组数。

接下来 t 行，每行一个 `int` 范围内的整数 n 。

输出格式

对于每组数据，输出转换之后的负二进制数。

样例输入1

```
3
-15
30000
-20000
```

样例输入1

```
110001
11011010101110000
1111011000100000
```

Hint

- 把十进制数转换成二进制数的方法是：除2取余，逆序排列。

```
#include<stdio.h>
int main(){
    printf("%d\n", -10 / -3);    //3
    printf("%d\n", -10 % -3);   //-1
    printf("%d\n", 10 % -3);    //1
    printf("%d\n", -10 % 3);    //-1
    return 0;
}
```

我们发现，取模运算的结果可能为负数，有什么办法能将负的余数变为正数呢？

- 余数公式是：被除数 ÷ 除数 = 商 余数
- 再提示就白给了(" $>$ " 目 " $<$)

AUTHOR: cxc

Problem I. xf 买彩票

题目描述

“世上本没有极光，深夜看基金的人多了，便有了极光。”—— xf

在经历了赛里斯基金市场的毒打之后， xf 在 yh 的建议下把投资方向转为买彩票，赛里斯国的彩票中奖号码有这样一个规律：号码中蕴含着某个能够被8整除的数， xf 已经选中了 t 个号码，你能帮他判断一下这些号码有中奖的可能吗？

Tips: 如果说一个数列 a_1, a_2, \dots, a_n 蕴含数列 $b_1, b_2, \dots, b_m (m \leq n)$, 则可以通过去除数列 a 中的某些元素而不破坏剩余元素的相对位置形成数列 b

输入格式

输入一个正整数 t , ($t \leq 10$), 接下来 t 行每行输入一个非负整数号码 n 。 ($n \leq 10^{100}$)

输出格式

一共 t 行，如果输入的第 i 个号码有中奖的可能，即蕴含某个能被8整除的数，则第 i 行输出`yes`，否则第 i 行输出`no`（输出不包括双引号）。

样例输入

```
10
0
8
24
3454
1545697
116
115
111111
159762
956452
```

样例输出

```
yes
yes
yes
yes
yes
yes
no
no
yes
yes
```

样例解释

第4个输入为数蕴含的344可以被8整除

第5、6个输入为数蕴含的16可以被8整除

第9个输入为数中蕴含的152可以被8整除

第十个输入为数中蕴含的952可以被8整除

Hint

- 做题前先看数据范围有助于选择合适的数据类型
- 如果说某个数可以被8整除，则意味着它除以8的余数为0
- 可以被8整除的数有某种特殊的性质
- 一位数中能被8整除的数有0、8
- 两位数中能被8整除的数有16、24、32、40、48、56.....
- 三位或位数更多的数若能被8整除，则其最后三位数组成的数字可以被8整除，如1919808的最后三位是808，而808能被8整除，所以1919808可以被8整除
- Think twice,code once

AUTHOR: 买基金血亏的 xf

Problem J. NAF

题目背景

小函有个室友叫小王，他是网络空间安全学院的。他们的作业题中有个特殊的二进制表示方法——NAF，小王不会做，所以请教小函编一个程序，将一个普通的二进制数转换为NAF的表示。但是小函中秋节玩着玩着就忘记了这件事。那就麻烦聪明的你帮帮小函给小王编下程序吧！

题目描述

NAF是一种整数的特殊二进制表示方法，它基于整数的BSD (Binary Signed Digit) 表示。

对于正整数 A ， $A = a_{n-1}2^{n-1} + \dots + a_12^1 + a_02^0$ (其中 $n = \lceil \log_2 K \rceil + 1$, $a_i \in \{-1, 0, 1\}$)为 A 的长度为 n 的带符号二进制表示。

而对正整数 A ，若 $(a_{n-1}, \dots, a_1, a_0)$ 为一种BSD表示，且**其中没有两个连续的 a_i 是非零的**，则其为**NAF表示**。

已知NAF表示有如下性质：

- 整数的NAF表示是唯一的
- NAF表示的长度至多比二进制表示的长度多一个比特位

现在给定若干个 `int` 范围内的整数，请你给出其NAF表示。

输入格式

第一行一个正整数 t ，表示有 t 组数据。 ($1 \leq t \leq 10^5$)

接下来 t 行，每行一个非负整数 k ，保证 k 在 `int` 范围内。

输出格式

输出一共 t 行，对于每组数据，输出一个**不带前导零**的带符号二进制串，并且该二进制串的每一位之间**由空格隔开**，代表 k 的NAF表示。

特殊地，如果 k 为 0，则输出 0。

样例输入

```
5
0
1
2
3
2147483647
```

样例输出

```
0
1
1 0
1 0 -1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1
```

样例解释

如，3的二进制表示为 11，而NAF表示为 1 0 -1

如，2147483647的二进制表示为 0111 1111 1111 1111 1111 1111 1111 1111，而NAF表示为 1 0 -1

Hint

注意非负整数无需考虑符号位问题

AUTHOR: yyh