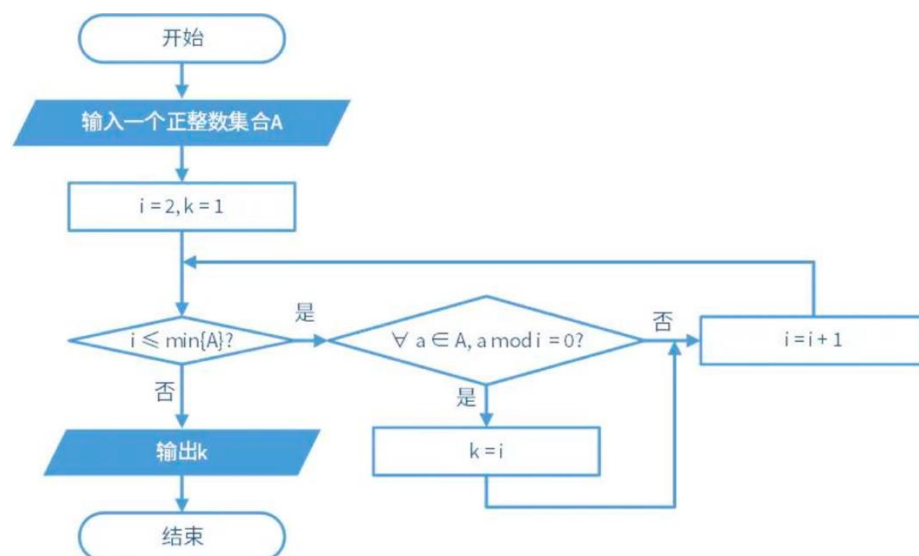


Lab 2

Task 1 最大公约数 GCD (A, B)

1.1 短除法

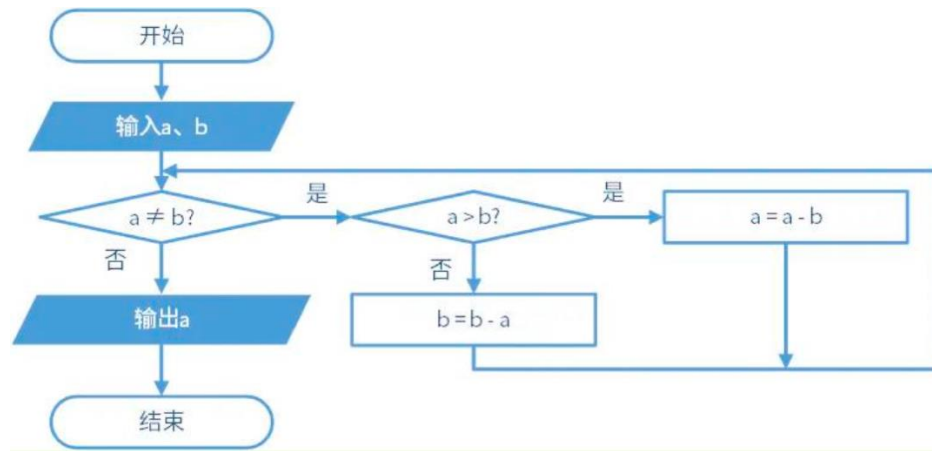
根据下述流程框图，实现 GCD (A, B)



```
int gcd1(int a, int b) {  
    int i = 2;  
    int k = 1;  
  
    while (i <= a && i <= b) {  
        // TODO  
    }  
    return k;  
}
```

1.2 更相减损术

根据下述流程框图，实现 GCD (A, B)

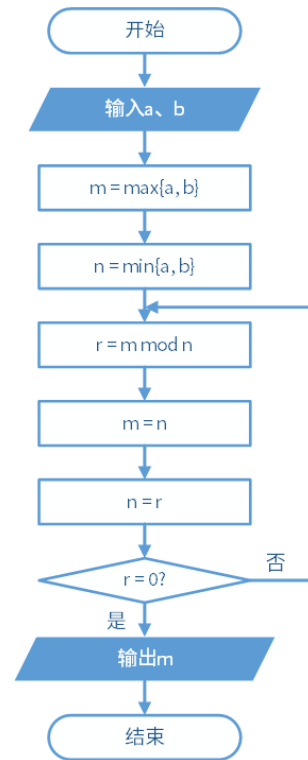


```
int gcd2(int a, int b) {  
    while (a!=b){  
        // TODO  
    }  
    return a;  
}
```

1.3 辗转相除法

根据下述流程，实现 GCD (A, B)

- 首先用两数中较大数除以较小数，求得数；
- 再用较小数和余数按上述操作进行相除；
- 直到余数为 0，此时的除数即为最大公约数。

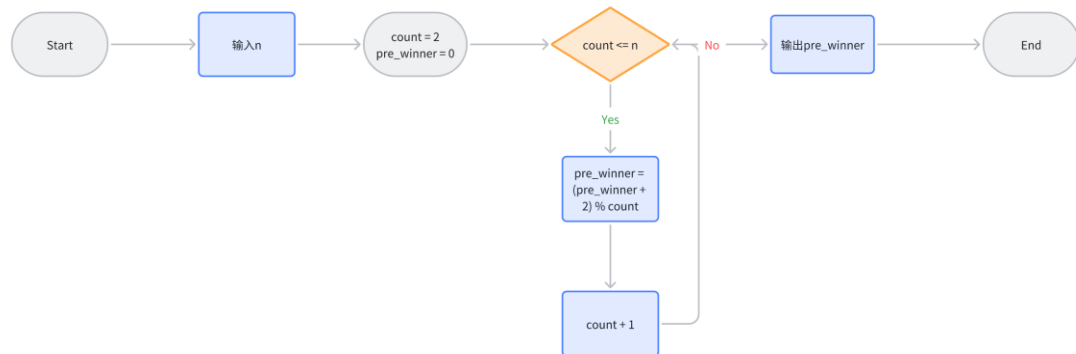


```
int gcd3(int a, int b) {  
    int m = 0;  
    int n = 0;  
    int r = a;  
  
    if(a > b){  
        // TODO  
    }  
    else {  
        // TODO  
    }  
  
    while(r != 0){  
        // TODO  
    }  
    return m;  
}
```

Task 2 – Homework: Josephus 问题

有 n 个人排成一圈，按顺时针方向依次编号 $0, 1, 2, \dots, n-1$ 。从编号为 0 的人开始顺时针“一二”报数，报到 2 的人退出圈子。这样不断循环下去，圈子里的人将不断减少，最终一定会剩下一个人。编写一段程序，输入 n ，输出最后剩下的人的编号。

例如，如果有 5 个人，那么依次是编号为 $1, 3, 0, 4$ 的退出，最后剩下编号为 2 的。



要求：

- (1) 实现源代码 (task2.c)
- (2) 提供实验报告，包括 $n=10\sim20$ 的结果统计表，以及对该流程图算法的理解。
- (3) 尝试使用数组/链表等进行实现，不需要按照流程图。 (Bonus)

思考：如果将报道 2 的人退出圈子，改为报到 m 的人退出圈子 (其中, $m < n$)，该如何修改上面的程序？

Task 3 – Homework: 闰年展示

输入 x, y 两个正整数 ($1582 \leq x < y \leq 3000$)，程序计算并输出 $[x, y]$ 区间中所有闰年年份数字，以及该区间内闰年的总数量。

提示：根据定义，

- 公元年份非 4 的倍数，为 365 天平年。
- 公元年份为 4 的倍数但非 100 的倍数，为 366 天闰年。
- 公元年份为 100 的倍数但非 400 的倍数（1700、1800、1900 年）为平年。
- 公元年份为 400 的倍数但非 4000 的倍数（1600 年及 2000 年）为闰年。
- 公元年份为 4000 的倍数为平年。

格里历闰年计算方法

年份	4的倍数但非整百数	整百数但非400的倍数	400的倍数但非4000的倍数	4000的倍数
结果	闰年	平年	闰年	平年

示例输入：

1989 2001

示例输出：

1992 1996 2000

3

要求：

1. 实现源代码（task3.c）；
2. **Bonus:** 先输出闰年总数，随后依次输出闰年年份数字。