约瑟夫环问题

限制

- 1000 ms
- 32768 KB

在学习了链表结构后，这一节我们需要用链表解决一个稍有改动的“约瑟夫环（Josephus problem）”问题：

计算理工学院有 N 个同学，围成了一个圆圈，每人被顺序地编了一个序号（分别为 $1, 2, 3...n$），从编号为 K 的人开始报 1，他之后（顺初始数字增长方向计算序号）的人报 2，以此类推，数到某一个数字 M 的人出列。出列同学的下一个人又从 1 开始继续报数，数到某一个数字 M 的人出列。不断重复这一过程，直到所有人都出列为止。

你需要根据同学人数 N 和给出的 K 和 M 计算出同学的正确出列顺序。

这一题的`main`函数已经帮你写好了，同时，已经帮你定义了一个节点的结构体类型、通过`circle_create`创建了一个循环链表。

现在请在`count_off`函数中根据传入的编号为 1 的节点 $head$、学生数 n、起始报数学生编号 k、数到出列的数字 m 实现报数的过程，按照题目要求进行输出。

输入格式

测评机会反复运行你的程序。每次程序运行时，输入为一行，包括三个被空格分隔开的符合描述的正整数 N、K 和 M（$1 \leq K \leq N \leq 1000$，$1 \leq M \leq 2000$）。

输出格式

输出为一行，包含 N 个整数，为依次顺序出列的学生编号，由空格分隔开。

样例输入 1

```
9 1 1
```

样例输出 1

```
1 2 3 4 5 6 7 8 9
```

样例输入 2

```
8 5 2
```

样例输出 2

```
6 8 2 4 7 3 1 5
```

提示信息

现在请在`count_off`函数的实现可以被拆分成以下三步进行思考：

- 通过使用 `head` 进入链表并使用 `->next` 若干次找到可以找到编号为 `k` （结构体中的 `data` 为 `k`）的人。
- 构建一个用于把 `n` 个人都出列的循环，每次循环中完成一个人出列的任务。
- 每一个让人出列的任务都从当前指向的人开始（第一次自然就是我们之前找到的那个编号为 `k` 的人），用一个循环往后数 `m` 次（也就是使用 `m` 次 `->next`）。这时候，我们指向的这个人是要出列的。让一个人出列可以通过三步走的方式。
  - 用一个单独声明的临时指针指向要出列的这个人
  - 将原本用 `->next` 指向要出列的这个人的元素指向要出列的这个人的下一个元素。
  - 使用临时指针执行 `free(temp)`。并将临时指针通过 `temp = NULL` 赋值为 `NULL`。（假设临时指针叫 `temp`）

更多提示：

- 要注意，第一个节点的前一个节点是最后一个节点喔。

- 要记住最后释放内存，争取不要出现内存泄露哦（如果没有处理好虽然不会扣分，但是还是建议你严格要求自己喔）。

请注意，输出请严格按照输出格式的要求进行，不要让你的程序输出任何多余的内容，否则测评机都会给出“运行结果错误”的提示。