约瑟夫问题实验设计报告

1. 问题描述

将问题要求分成四个执行阶段

输入阶段：

从键盘输入总人数N（N小于50），接着输入N个人的信息，包括编号（整型）、姓名、性别和年龄，并建立一个带头节点的循环链表存储这些信息。

输出阶段：

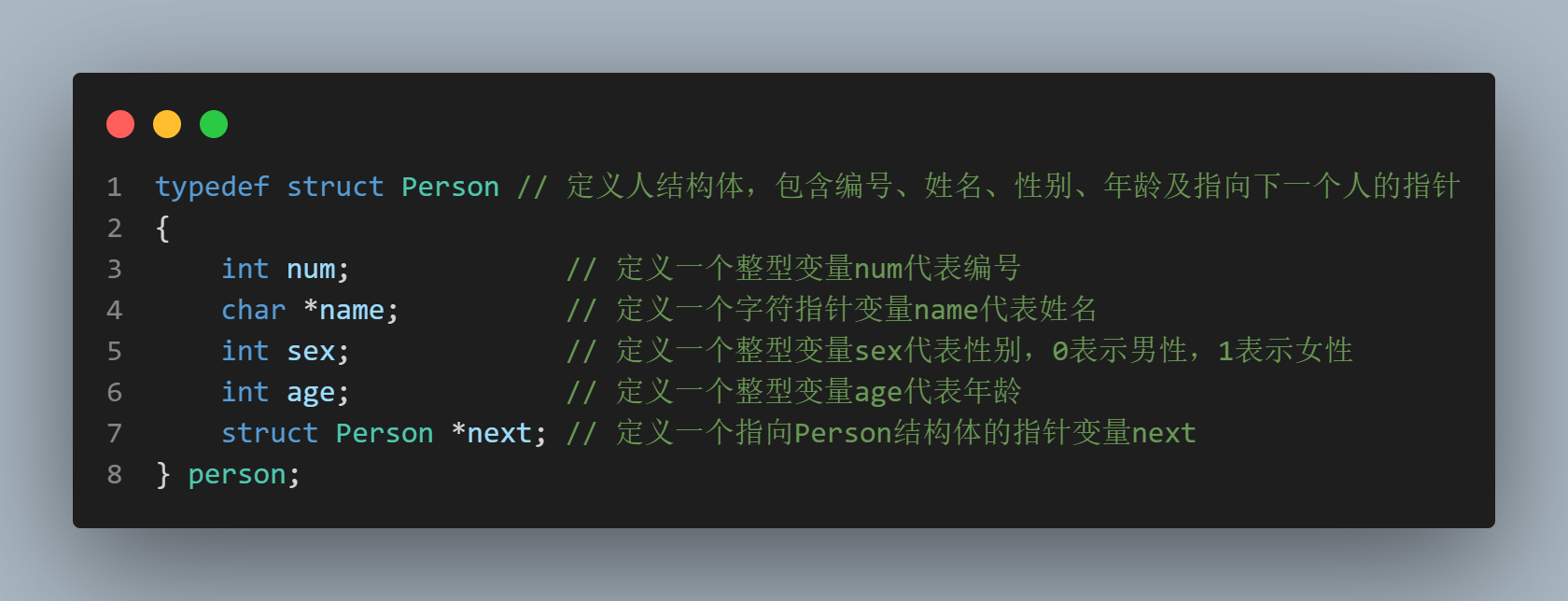
输出当前循环链表中所有结点的值以验证链表是否正确构建。

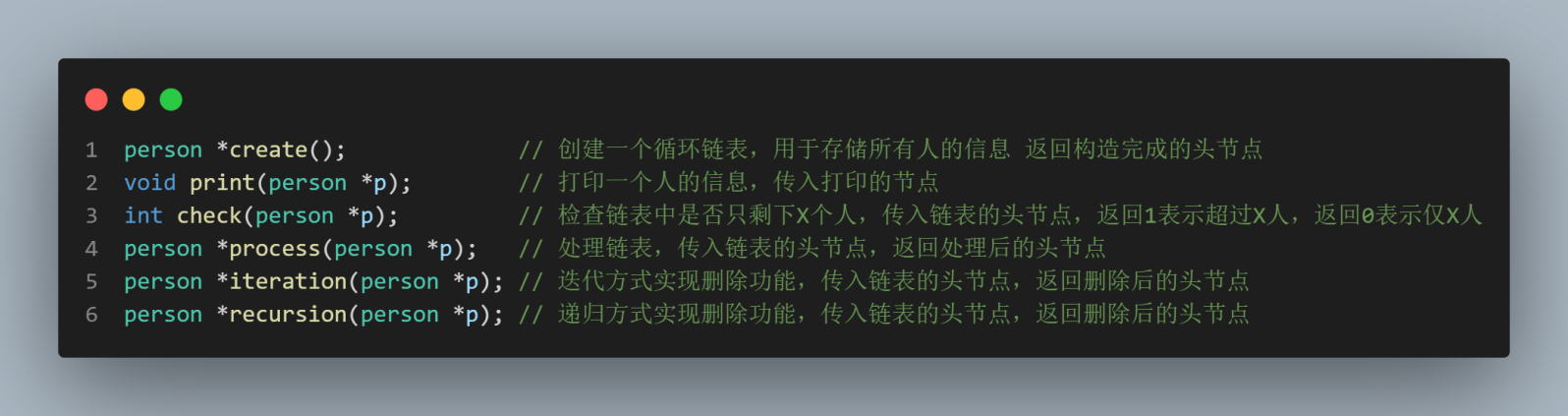
处理阶段：

输入开始报数的人的编号S、报数的间隔M和最终要剩余的人数X；在循环链表中找到编号为S的结点作为起始点；从该结点开始，按照间隔M进行报数，每报到M时删除该结点，并输出被删除结点的编号；删除操作继续，从被删除结点的下一个结点开始，重复报数和删除过程，直到链表中只剩X个人。

最终输出阶段：

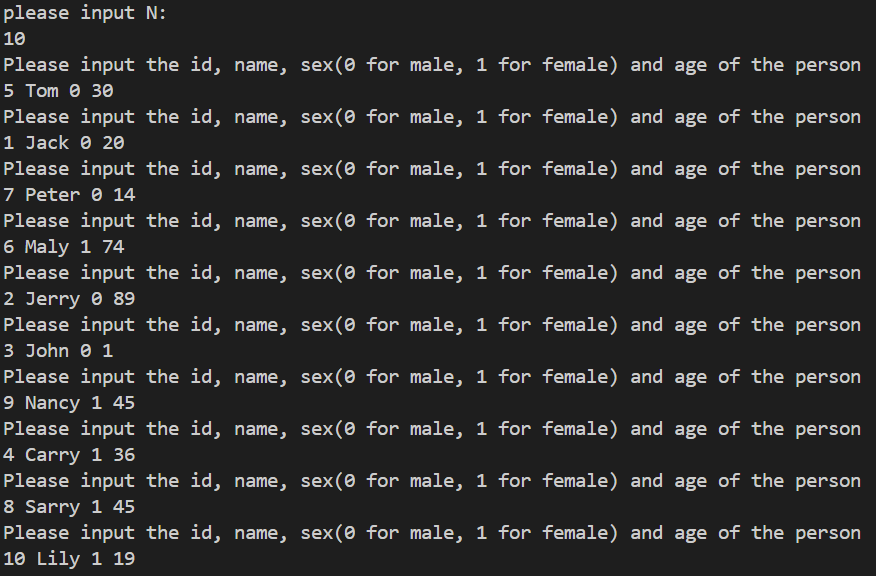
输出最后剩余的X个人的详细信息，包括编号、姓名、性别和年龄。

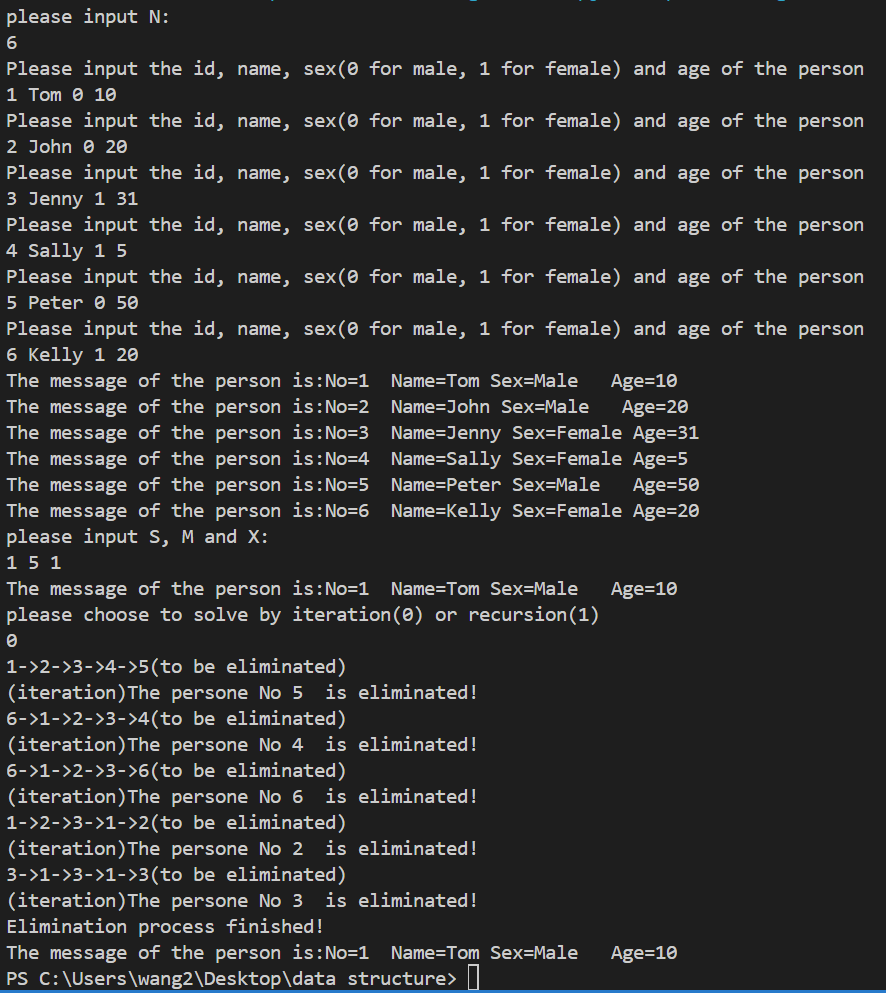
1. 设计思路
2. 定义链表节点结构：首先定义一个结构体包含一个人的编号、姓名、性别、年龄以及指向下一个人的指针
3. 创建与构造链表：编写一个函数先创建一个空的链表头节点并接收用户输入的人数N，然后循环N次，每次循环接收一个人的编号、姓名、性别和年龄，并创建一个新节点将其插入到链表中。最后将最后的节点指向头节点形成循环链表
4. 输出链表：编写一个函数来输出一个节点储存的信息
5. 实现删除和报数的逻辑：首先，编写一个函数来找到链表中编号为S的节点。然后，从该节点开始，按照间隔M进行报数并删除节点，重复执行直到剩余X个节点。用户可自行选择用迭代的方法还是递归的方法解决问题
6. 输出最终结果：遍历并输出链表中剩余的X个节点的详细信息。
7. 代码说明
8. 结构体定义
9. 函数定义和功能

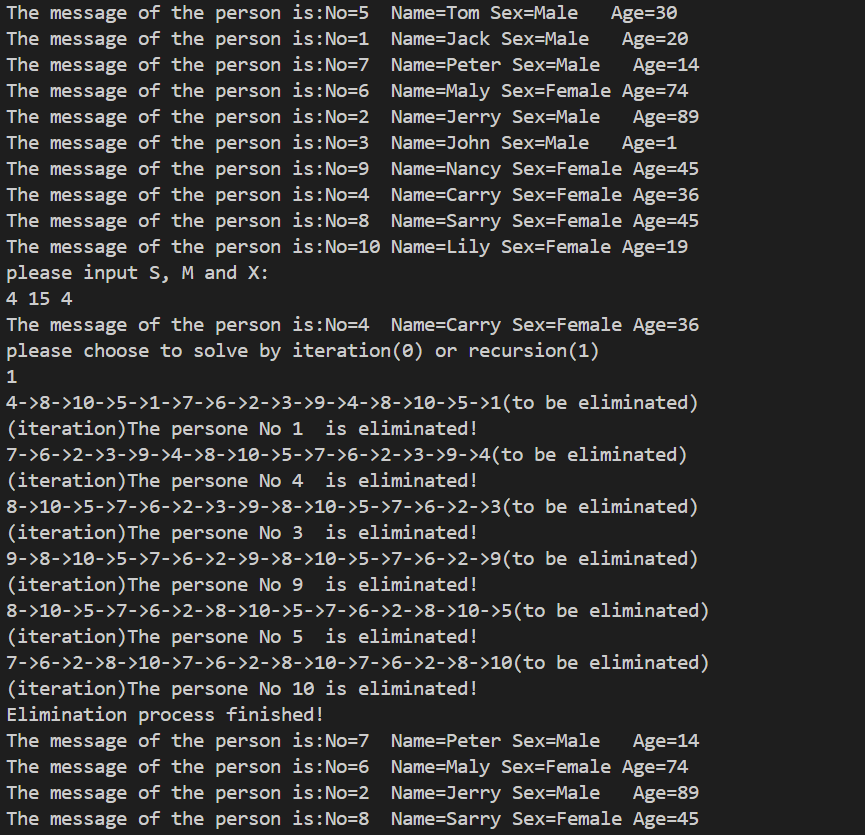


1. 函数复杂度分析

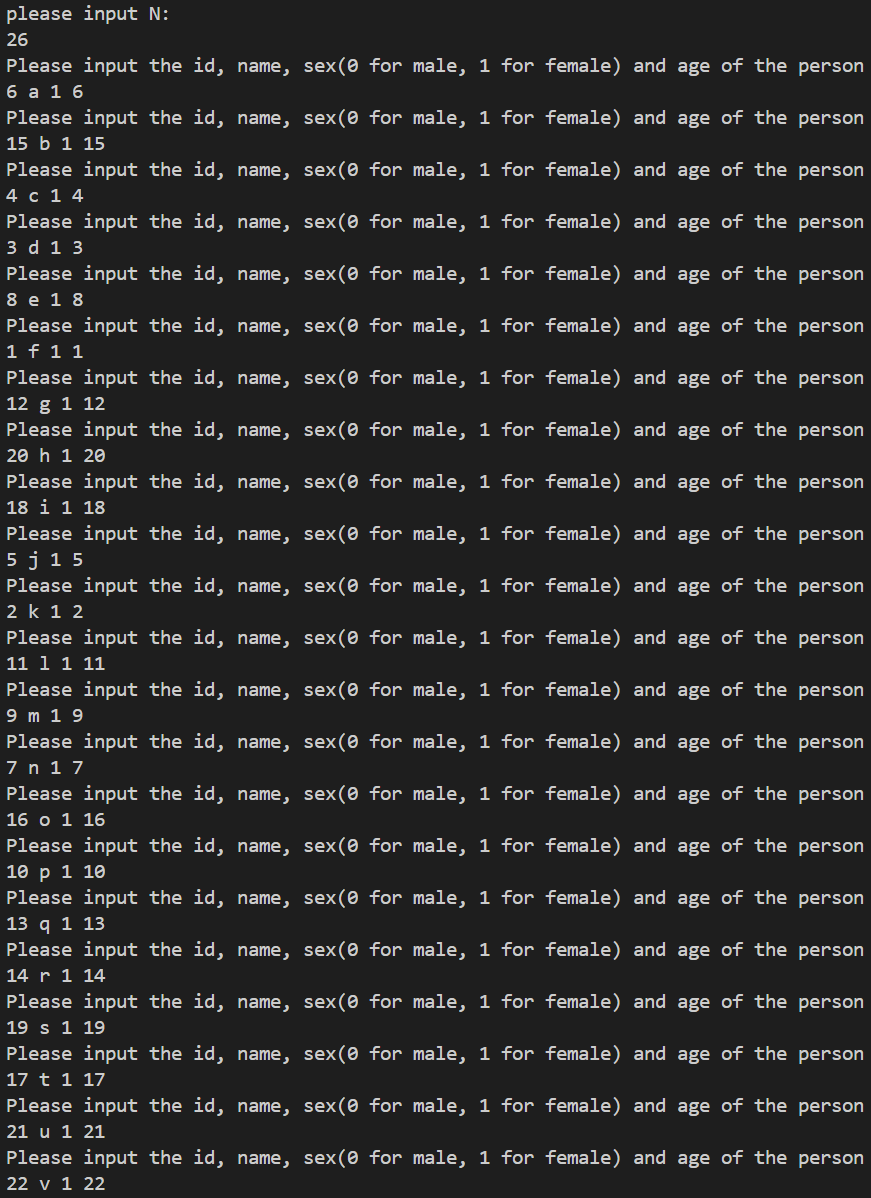
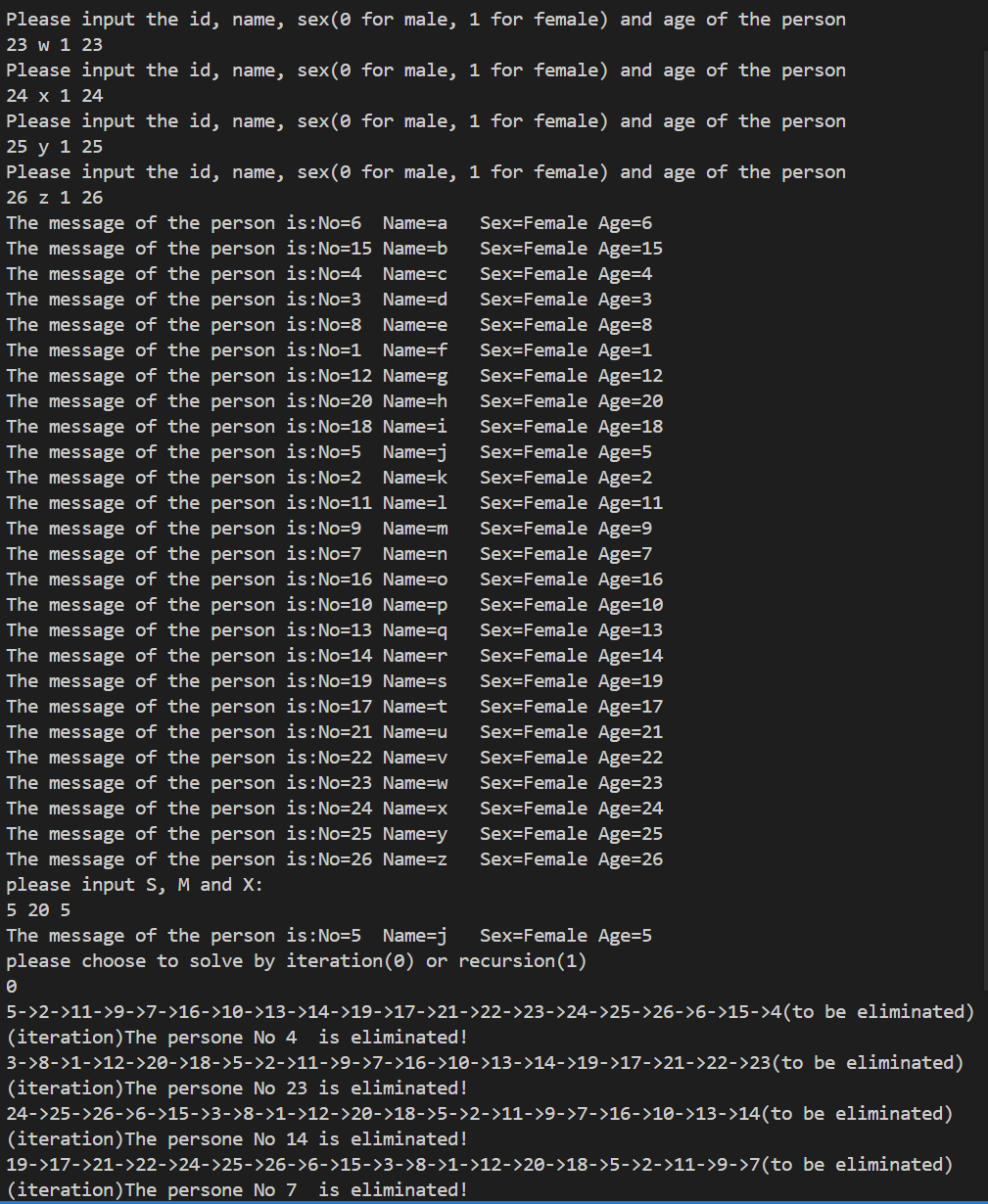
|  |  |  |
| --- | --- | --- |
| 函数 | 时间复杂度 | 空间复杂度 |
| person \*create() | O(n) | O(n) |
| void print(person \*p) | O(1) | O(1) |
| int check(person \*p) | O(n) | O(1) |
| person \*process(person \*p) | O(n) | O(1) |
| person \*iteration(person \*p) | O(n2) | O(1) |
| person \*recursion(person \*p) | O(n2) | O(n) |

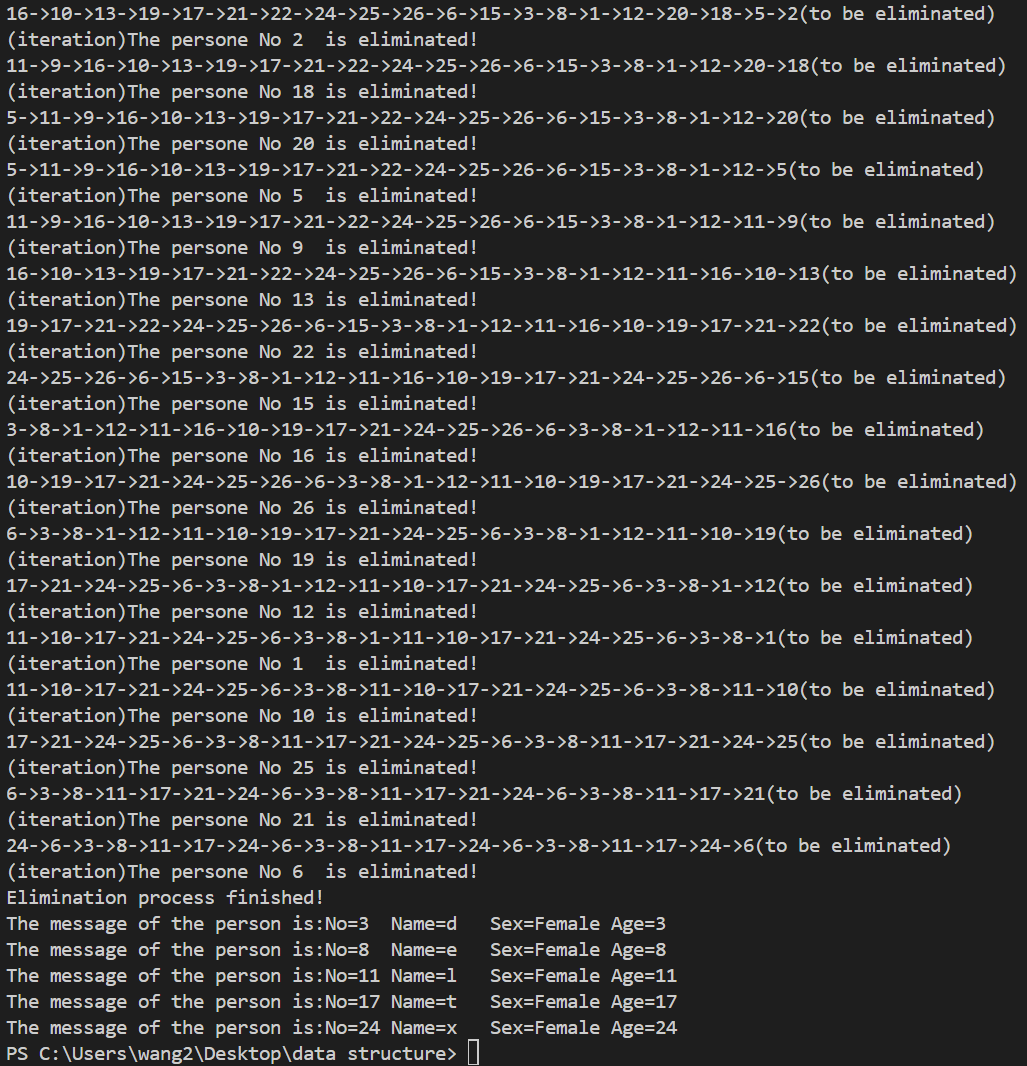


1. 运行结果截图



运行结果1：样例输出 运行结果2：M>N的情况





运行结果3：较大的样例