

实验一 链表的实现与应用

实验目的

1. 熟悉 C/C++ 集成开发环境，能够利用 C/C++ 语言实现基本算法
2. 掌握链表的存储结构，实现链表的基本操作
3. 合理设计运用数据结构，能够编写高效率算法解决应用问题

实验内容

题 1：示例程序调试

根据示例程序理解单链表的数据类型定义和基本操作的实现，熟练在集成开发环境（推荐使用 Visual Studio Code）中的编辑、编译、运行和调试方法。示例程序包含头文件 linklist.h，实现文件 linklist.c，以及主程序 test_linklist.c，其中 linklist.c 中部分基本操作函数的实现可能存在若干错误，请尝试排除，并编译运行得到以下正确结果：

```
1. create an empty linked list...success!
2. insert elements {'q','w','e','r','t','y'} in turn: qwerty
    insert q at 1 success!
    insert w at 2 success!
    insert e at 3 success!
    insert r at 4 success!
    insert t at 5 success!
    insert y at 6 success!
3. traverse the linked list: q w e r t y
4. the list has a length of 6, i.e., not empty.
5. search for the element at position: 5
    the 5th element in the list is: t
6. insert 'f' at position 4...success!
    now the linked list is: q w e f r t y
7. the element for search is: y
    the element y in the list is at position 7
8. delete the 3rd element in the list...success!
    now the linked list is: q w f r t y
9. destroy the linked List...success!
```

图 1-1 题 1 正确运行结果

题 2: 约瑟夫问题求解

约瑟夫 (Josephus) 问题又称为约瑟夫环, 其一种游戏形式的描述是: 编号为 $1, 2, \dots, n$ 的 n 个人按顺时针方向围坐一圈做游戏, 每人持有一个密码 (正整数)。一开始任选一个正整数作为报数上限值 m , 从第一个人开始按顺时针方向自 1 开始报数, 报到 m 时停止报数。报 m 的人出列, 将他的密码作为新的 m 值, 从他在顺时针方向的下一个人开始重新从 1 报数, 这个过程沿着圆圈一直进行下去, 直至圈内最终仅剩下 1 个人为止, 此人即胜出者。

试设计一个程序实现约瑟夫问题求解算法, 并按出列顺序打印各人编号及胜出者。下图给出当 $n=7$, $m=20$, 个人密码分别为 $\{3, 1, 7, 2, 4, 8, 4\}$ 时该算法正确的输出序列:

```
game g1 started...
the initial list is:
(1,3) (2,1) (3,7) (4,2) (5,4) (6,8) (7,4)
the losers are in turn:
6 1 4 7 2 3
the winner is: 5.
```

图 1-2 题 2 示例数据运行结果

实验要求

1. 认真阅读并在 6 学时内完成本次实验的所有内容
2. 灵活运用课程所学数据结构, 提前做好问题分析与数据建模, 完成详细代码设计和上机调试
3. 程序须从指定文件 game.txt 导入信息, 该文件格式及相关说明如下:

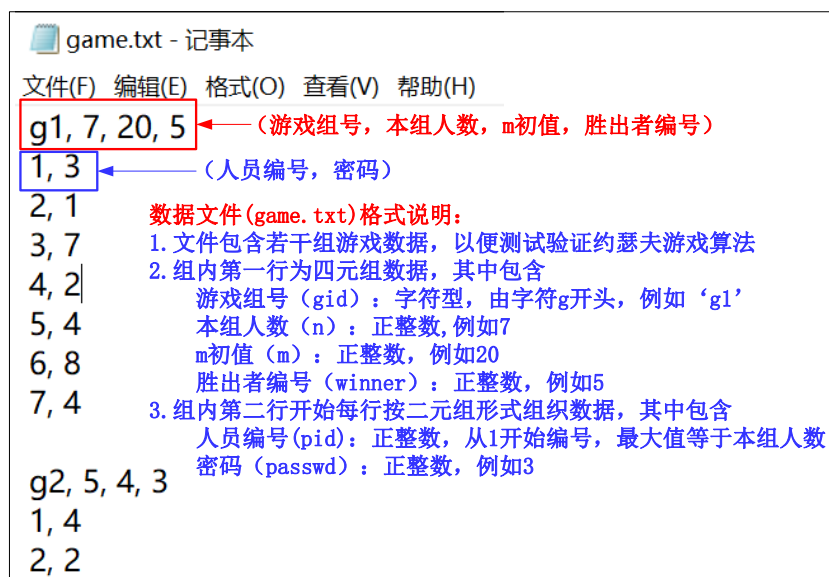


图 1-3 数据文件 game.txt 的格式说明

4. 程序应支持从 game.txt 按指定格式依次读取并运行多组游戏的能力, 且可以统计出算法求解正确的比率, 即: 正确求解次数/总组数
5. 针对每一组游戏, 程序至少应具备按出列顺序打印人员编号和胜出者编号等基本功能
6. 每次上机结束后请自行使用 U 盘保存好源程序和可执行文件, 以备工作接续和检查之需
7. 建议拍照保留程序运行的重要过程和结果, 以备撰写实验报告时使用