



北京邮电大学

Beijing University of Posts and Telecommunications

北京邮电大学计算机学院

《数据结构》实验设计报告

学 号：2020211435

姓 名：何家豪

班 级：2020211308

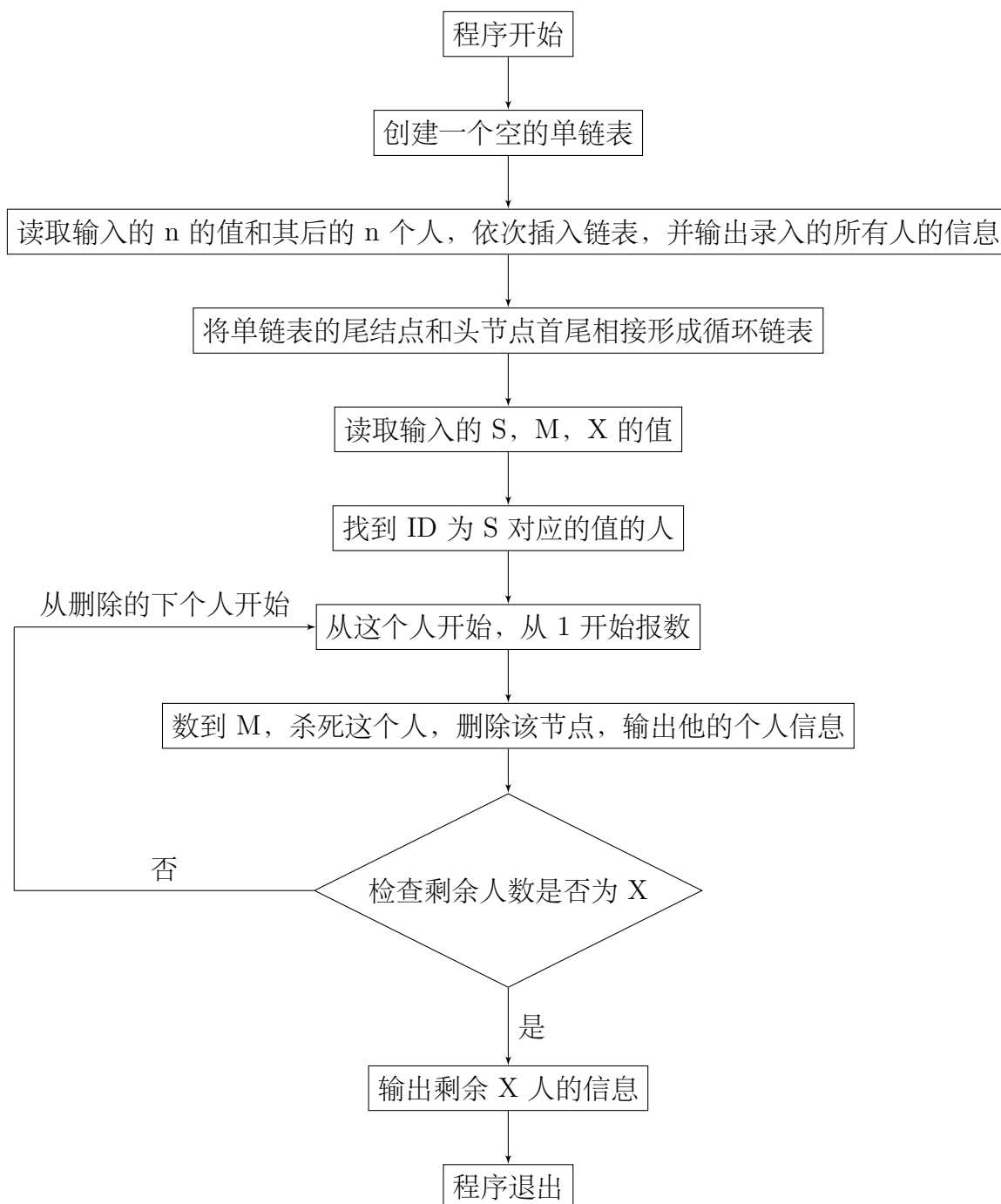
授课教师：张海旻

完成日期：2021 年 10 月 19 日

目录

| | |
|------------------------|---|
| 1 设计思路 | 1 |
| 2 代码说明 | 2 |
| 2.1 常量定义 | 2 |
| 2.2 结构体定义 | 2 |
| 2.2.1 人结构体 | 2 |
| 2.2.2 链表结点结构体 | 2 |
| 2.3 函数设计及其功能 | 3 |
| 2.3.1 initList() | 3 |
| 3 运行结果 | 4 |
| 3.1 运行环境 | 4 |
| 3.1.1 硬件信息及系统版本 | 4 |
| 3.1.2 GCC 编译器版本 | 4 |
| 3.1.3 编译指令 | 5 |

第一部分 设计思路



第二部分 代码说明

2.1 常量定义

```
1 #define NAME_LENGTH 21 // 人名的最大长度
2 #define MAX_PEOPLE 50 // 最大人数
```

- NAME_LENGTH 定义了人名的最大长度，考虑到字符串最后的结束符，实际人名最长长度为 20。
- MAX_PEOPLE 定义了最多的参与人数：50 人。

2.2 结构体定义

2.2.1 人结构体

```
1 // 定义人结构体
2 typedef struct Person
3 {
4     int id;           // 编号
5     char name[NAME_LENGTH]; // 姓名
6     int gender;       // 性别
7     int age;          // 年龄
8 } Person;
```

在人结构体中，有人的编号，姓名，性别和年龄。

2.2.2 链表结点结构体

```
1 // 定义链表结点结构体
2 typedef struct ListNode
```

```
3 {  
4     struct ListNode *next; // 指向下一节点  
5     Person *person;        // 当前人  
6 } Node, *LinkedList;
```

在结点的结构体中有指向下一节点的指针域和指向该节点代表的人的结构体指针。这里选择将人的指针作为数据存在结点中，而不是人结构体本身，好处是可以减少在参数传递时的负担。

2.3 函数设计及其功能

2.3.1 initList()

```
1 // 创建一个带有头节点的空链表  
2 int initList(ListNode *L)  
3 {  
4     ListNode newList = (Node *)malloc(sizeof(Node)); //  
5     // 为头节点分配空间  
6     if (!newList) // 如果空间分配失败  
7     {  
8         printf("initList: 创建新链表失败。\\n");  
9         exit(1);  
10    }  
11    // 将表头的指针域和data域置空  
12    newList->next = NULL;  
13    newList->person = NULL;  
14    *L = newList;  
15    return 0;  
16 }
```

该函数的功能是创建一个仅有头节点的空链表。如果在为头节点分配内存时出现错误，将会抛出错误代码 1，并退出程序。若创建成功，将会把头节点的指针域和数据域都设置为 NULL。头结点将是唯一一个数据域为 NULL 的结点，这也可以作为后续判断是否为头结点的判断条件。

第三部分 运行结果

3.1 运行环境

3.1.1 硬件信息及系统版本



3.1.2 GCC 编译器版本

```
1 (base) jasonhe@JasondeMacBook-Pro ~ % gcc --version
2 Configured with: --prefix=/Applications/Xcode.app/Contents/
   Developer/usr --with-gxx-include-dir=/Applications/
   Xcode.app/Contents/Developer/Platforms/MacOSX.
   platform/Developer/SDKs/MacOSX.sdk/usr/include/c
   ++/4.2.1
3 Apple clang version 13.0.0 (clang-1300.0.29.3)
4 Target: arm64-apple-darwin20.6.0
5 Thread model: posix
```

```
6 | InstalledDir: /Applications/Xcode.app/Contents/Developer/  
   | Toolchains/XcodeDefault.xctoolchain/usr/bin
```

3.1.3 编译指令

本部分的运行结果均是在以下编译指令编译完成，运行 JosephRing 可执行文件完成的。

```
1 | (base) jasonhe@JasondeMacBook-Pro ~ % gcc JosephRing.c -o  
   | JosephRing
```