**计算机科学与工程学院实验报告1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验课程名称** | | **实验一 线性表应用类实验**  **——约瑟夫环问题** | | | **实验成绩** |  |
| **专业** | **计算机科学与技术** | | **班级** | **2206** | **指导教师签字** |  |
| **姓名** | **李昕鸿** | | **姓名** | **黄璐媛** | **姓名** |  |
| **学号** | **20225868** | | **学号** | **20225956** | **学号** |  |
| **附录实验程序及结果：**  **实验程序：**    **结果：**    **源代码：**  #include<stdio.h>  #include<stdlib.h>  #include<iostream>  using namespace **std**;  typedef struct **node**  {       int data;       struct **node**\* next;       struct **node**\* next1;  }**Node**;  void **ysflb**(int N, int M)  {  **Node**\* head = **NULL**,\*p = **NULL**, \* r = **NULL**;       head = (**Node**\*)**malloc**(sizeof(**Node**));       if (**NULL** == head)       {            cout **<<** "Failed!";            return;       }       head->data = 1;       head->next = **NULL**;       head->next1 = **NULL**;       p = head;  *//尾插法创建链表*       for (int i = 2;i <= N;i++)       {            r = (**Node**\*)**malloc**(sizeof(**Node**));            r->data = i;            r->next = **NULL**;            r->next1 = p;  *//插入结点*            p->next = r;            p = r;       }  *//创建循环链表*       head->next1 = p;       p->next = head;       p = head;  *//约瑟夫环的模拟*       int times = 1;       while (p->next != p) *//只有一个元素*       {            if (times % 2 == 1)            {                 for (int i = 1;i < M;i++)                 {                      r = p; *//保留出局的前一个结点*                      p = p->next; *//p指向的是要出局的这个结点，需要保留前一个结点*                 }  *// 输出*                 cout **<<** p->data **<<** " ";                 r->next = p->next;                 p->next->next1 = r;                 p = p->next1; *//更新p重新进行报数*            }            else {                 for (int i = 1;i < M;i++)                 {                      r = p; *//保留出局的前一个结点*                      p = p->next1; *//p指向的是要出局的这个结点，需要保留前一个结点*                 }  *// 输出*                 cout **<<** p->data **<<** " ";                 r->next1 = p->next1;                 p->next1->next = r;                 p = p->next; *//更新p重新进行报数*            }            times++;       }       cout **<<** p->data;  }  int **main**()  {       int N, M;       cin **>>** N **>>** M;  **ysflb**(N, M);       return 0;  } | | | | | | |
| **实验课程总结**  通过本次实验课程内容的学习，我了解了有关线性表数据结构的相关问题。主要学习的是有关链表的使用，包含双向链表和循环链表。在初步编写时遇到了一些困难，如分奇偶次的时候指针的移动方向等。本次实验内容提升了我对线性表相关知识的了解，加强链表的应用。 | | | | | | |

**批改时间：**