编号：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 总评 | 教师签名 |
| 成绩 |  |  |  |  |  |  |  |  |  |  |  |  |

武汉大学计算机学院

课程实验(设计)报告

专业(班)： 计算机科学与技术1班

学 号： 2015301040209

姓 名： 张相宾

课程名称： 操作系统设计

任课教师： 宋伟

2017年 11 月 20 日

**磁盘存储空间的分配和回收**

1. **实习题目**
2. 实习内容

模拟磁盘空闲空间的表示方法，以及模拟实现磁盘空间的分配和回收。

1. 实习目的

磁盘初始化时把磁盘存储空间分成许多块（扇区），这些空间可以被多个用户共享。用户作业在执行期间常常要在磁盘上建立文件或把已经建立在磁盘上的文件删去，这就涉及到磁盘存储空间的分配和回收。一个文件存放到磁盘上，可以组织成顺序文件（连续文件）、链接文件（串联文件）、索引文件等，因此，磁盘存储空间的分配有两种方式，一种是分配连续的存储空间，另一种是可以分配不连续的存储空间。怎样有效地管理磁盘存储空间是操作系统应解决的一个重要问题，通过本实习使学生掌握磁盘存储空间的分配和回收算法。

1. 实习题目

第二题：

用位示图管理磁盘存储空间

(1) 为了提高磁盘存储空间的利用率，可在磁盘上组织成链接文件、索引文件，这类文件可以把逻辑记录存放在不连续的存储空间。为了表示哪些磁盘空间已被占用，哪些磁盘空间是空闲的，可用位示图来指出。位示图由若干字节构成，每一位与磁盘上的一块对应，“1”状态表示相应块已占用，“0”状态表示该块为空闲。位示图的形式与实习二中的位示图一样，但要注意，对于主存储空间和磁盘存储空间应该用不同的位示图来管理，绝不可混用。

(2) 申请一块磁盘空间时，由分配程序查位示图，找出一个为“0”的位，计算出这一位对应块的磁盘物理地址，且把该位置成占用状态“1”。假设现在有一个盘组共8个柱面，每个柱面有2个磁道（盘面），每个磁道分成4个物理记录。那么，当在位示图中找到某一字节的某一位为“0”时，这个空闲块对应的磁盘物理地址为：

柱面号=字节号

磁道号= 位数 / 4

物理记录号= 位数 % 4

(3) 归还一块磁盘空间时，由回收程序根据归还的磁盘物理地址计算出归还块在位示图中的对应位，把该位置成“0”。按照（2）中假设的盘组，归还块在位示图中的位置计算如下：

字节号=柱面号

位数=磁道号4+物理记录号

(4) 设计申请磁盘空间和归还磁盘空间的程序。

要求能接受来自键盘的空间申请及释放请求，要求能显示或打印程序运行前和运行后的位示图；分配时把分配到的磁盘空间的物理地址显示或打印出来，归还时把归还块对应于位示图的字节号和位数显示或打印出来。

1. **实习内容及设计思想**
2. 设计思路

位示图管理，按块划分，支持不连续分配及回收

1. 主要数据结构

Class task{ //作业

Bool request(); //申请磁盘空间

Bool printAdd(); //打印物理地址

Void restore(); //回收分配到的磁盘空间

Int id; //作业名

Int size; //作业所需大小

phyAddr addr[MAX]; //存储物理地址格式

}

Void printMap(); //打印位示图

Int getFreeBlock(); //得到位示图看空闲磁盘块数

1. 主要代码结构

a.请求

cout << "请输入作业需要的内存大小（kb）：";

cin >> size;

freeBlock = getFreeBlock();

if (size <= freeBlock) {

pro.push\_back(task(tId, size));

iter = pro.end();

iter--;

iter->request();

iter->printAdd();

}

b.回收

cout << "请输入要回收作业号：";

cin >> tId;

for (iter = pro.begin(); iter != pro.end(); iter++) {

if (tId == iter->id) {

judge = true;

break;

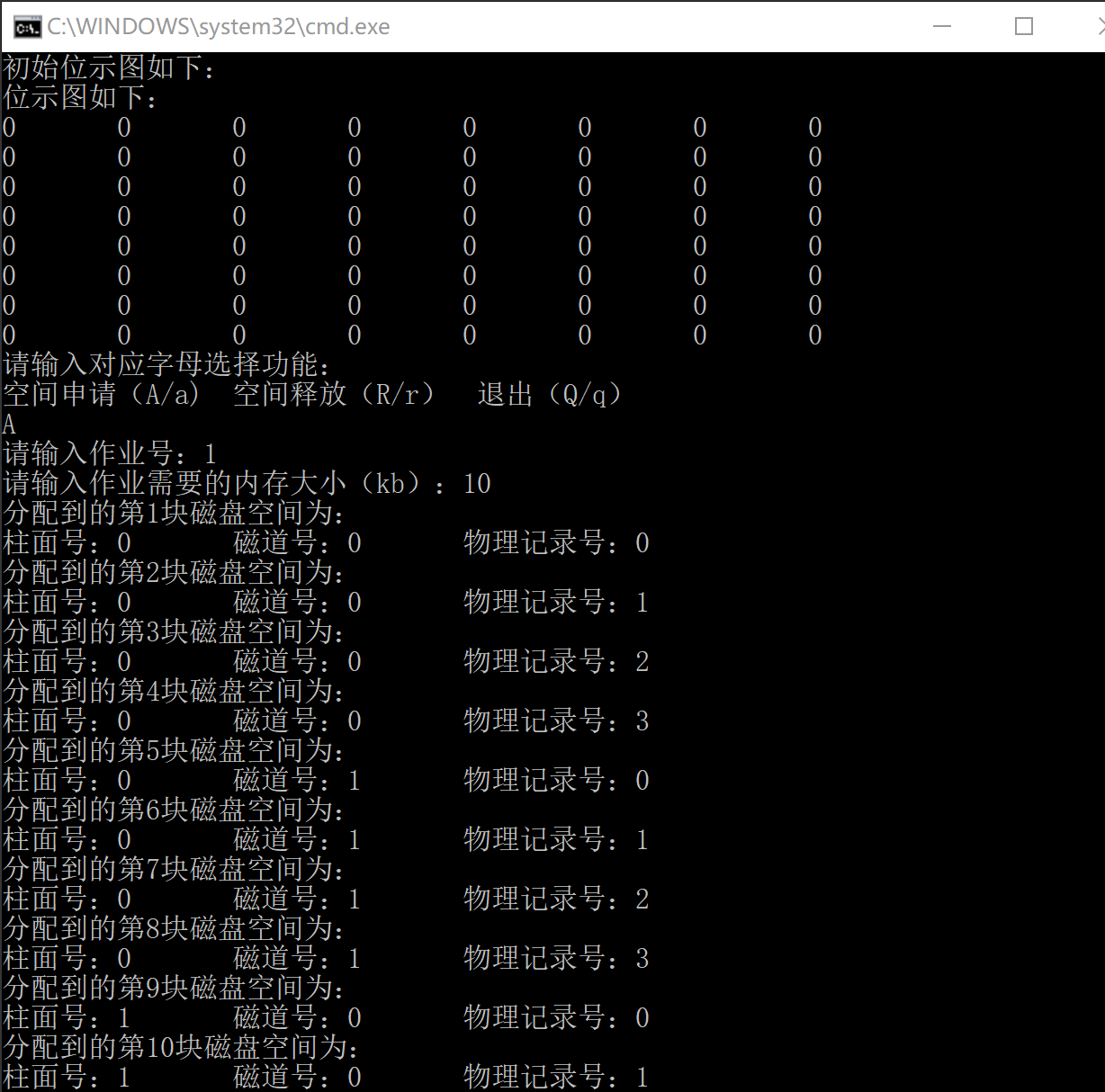
}

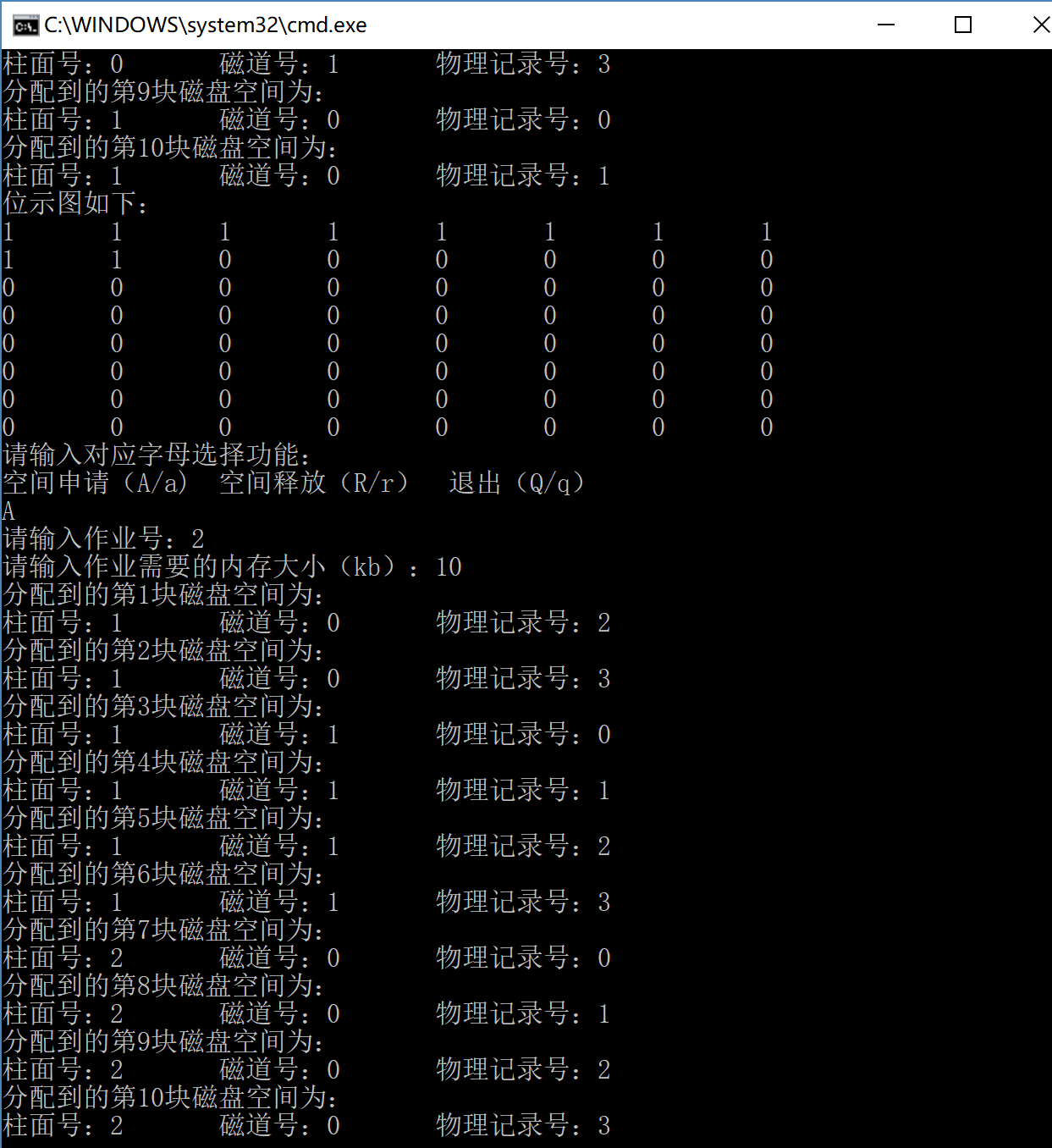
}

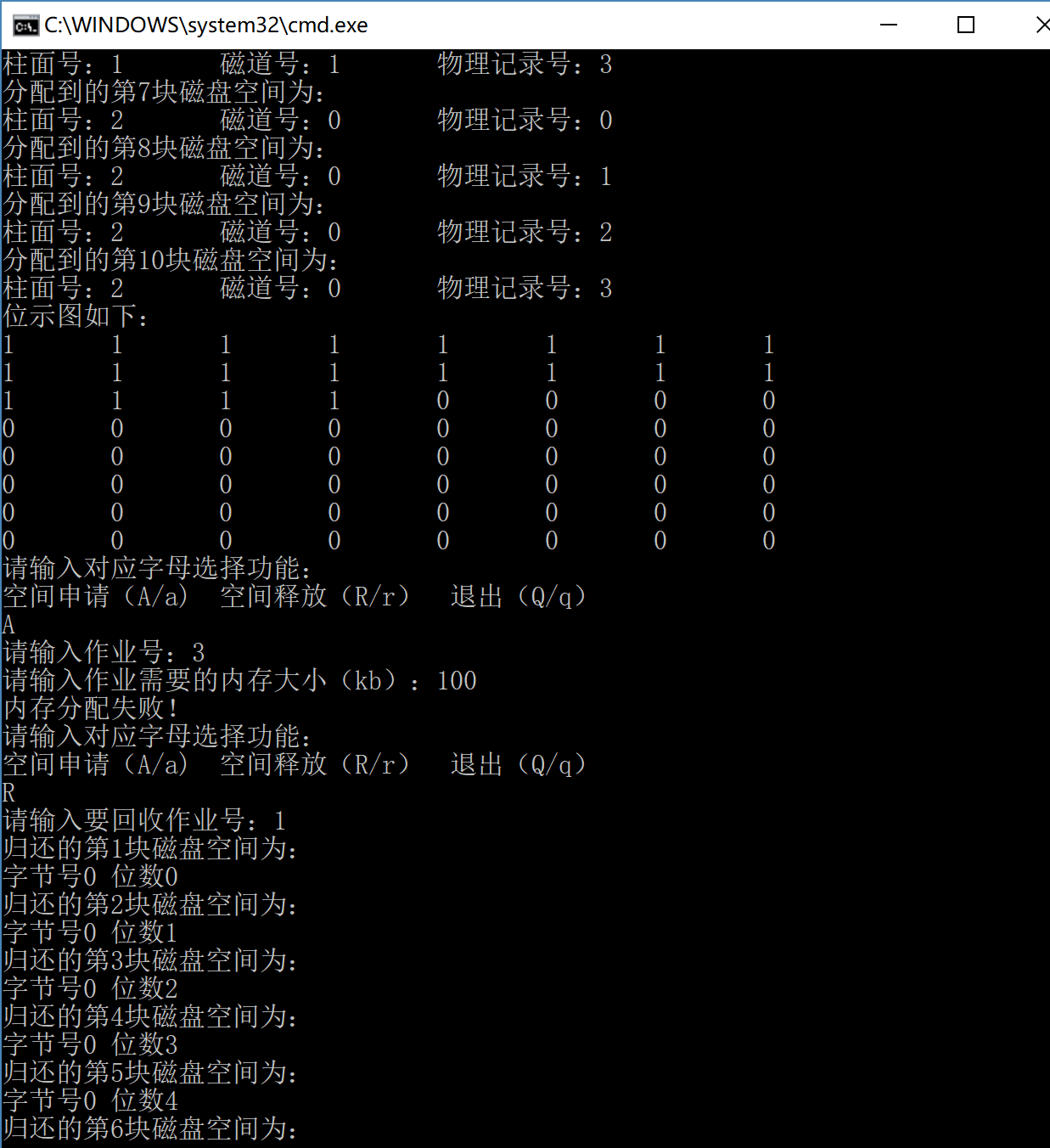
1. 代码段分析
2. **上机实习所用平台及相关软件**
3. **调试过程**
4. 测试数据设计

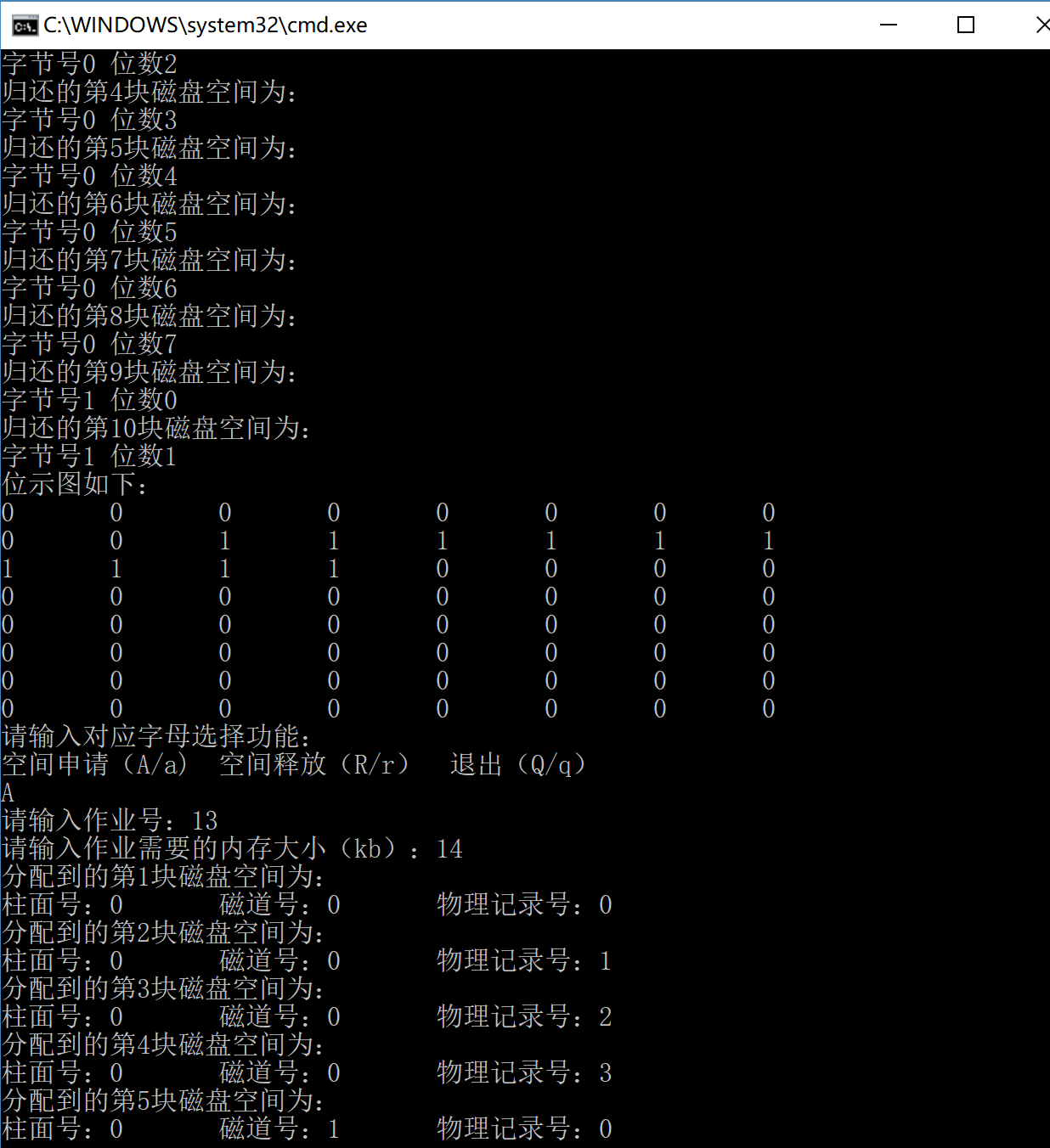
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 执行序列 | 作业号 | 作业大小 | 命令 | 预测结果 |
| 1 | 1 | 10 | 请求分配 | 成功 |
| 2 | 2 | 10 | 请求分配 | 成功 |
| 3 | 3 | 100 | 请求分配 | 磁盘空间不足，无法分配 |
| 4 | 1 | ~ | 空间释放 | 成功 |
| 5 | 13 | 14 | 请求分配 | 成功，分配空间不连续 |
| 6 | 2 | ~ | 空间释放 | 成功 |
| 7 | 13 | ~ | 空间释放 | 成功释放不连续空间 |

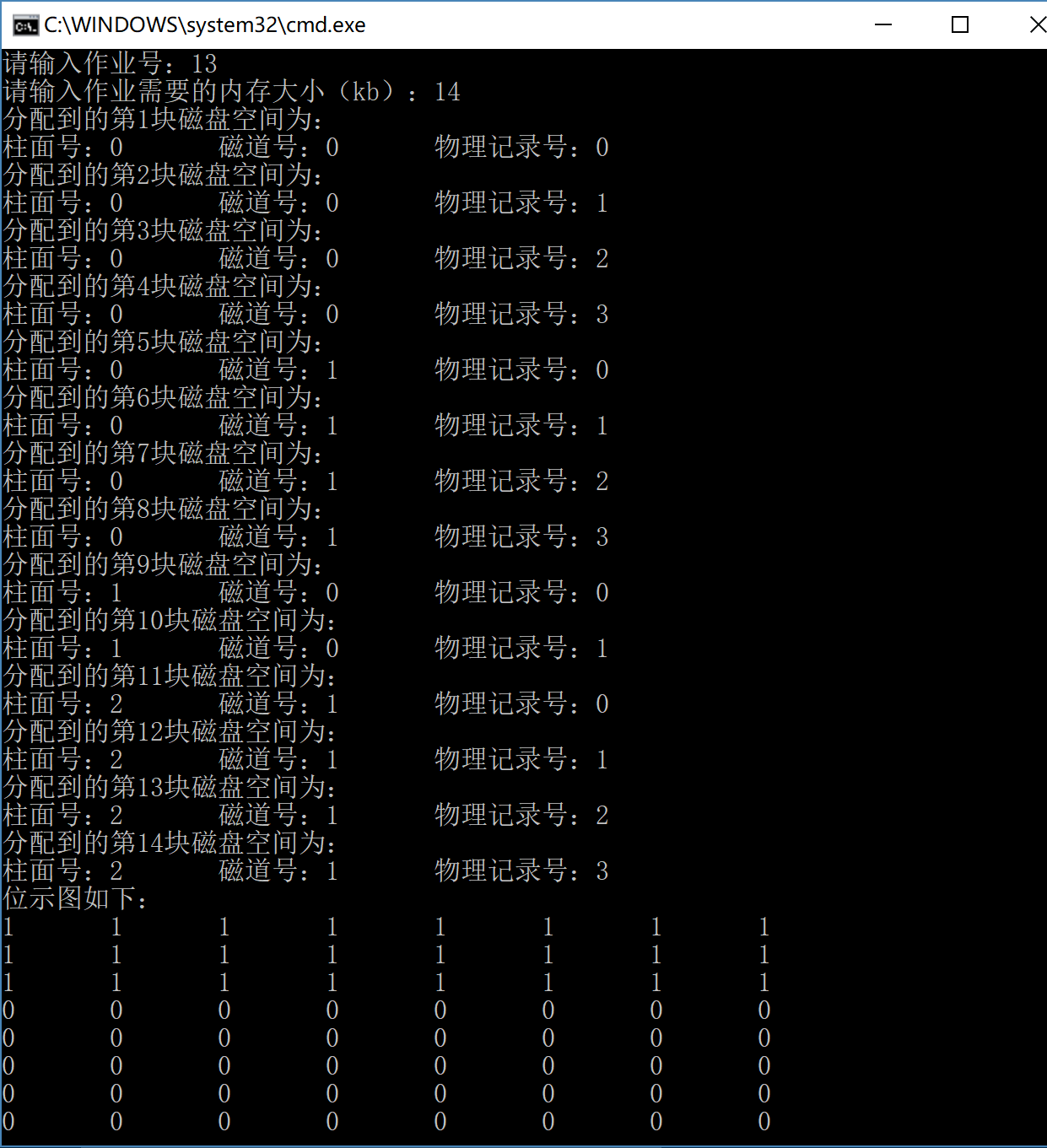
1. 测试结果分析

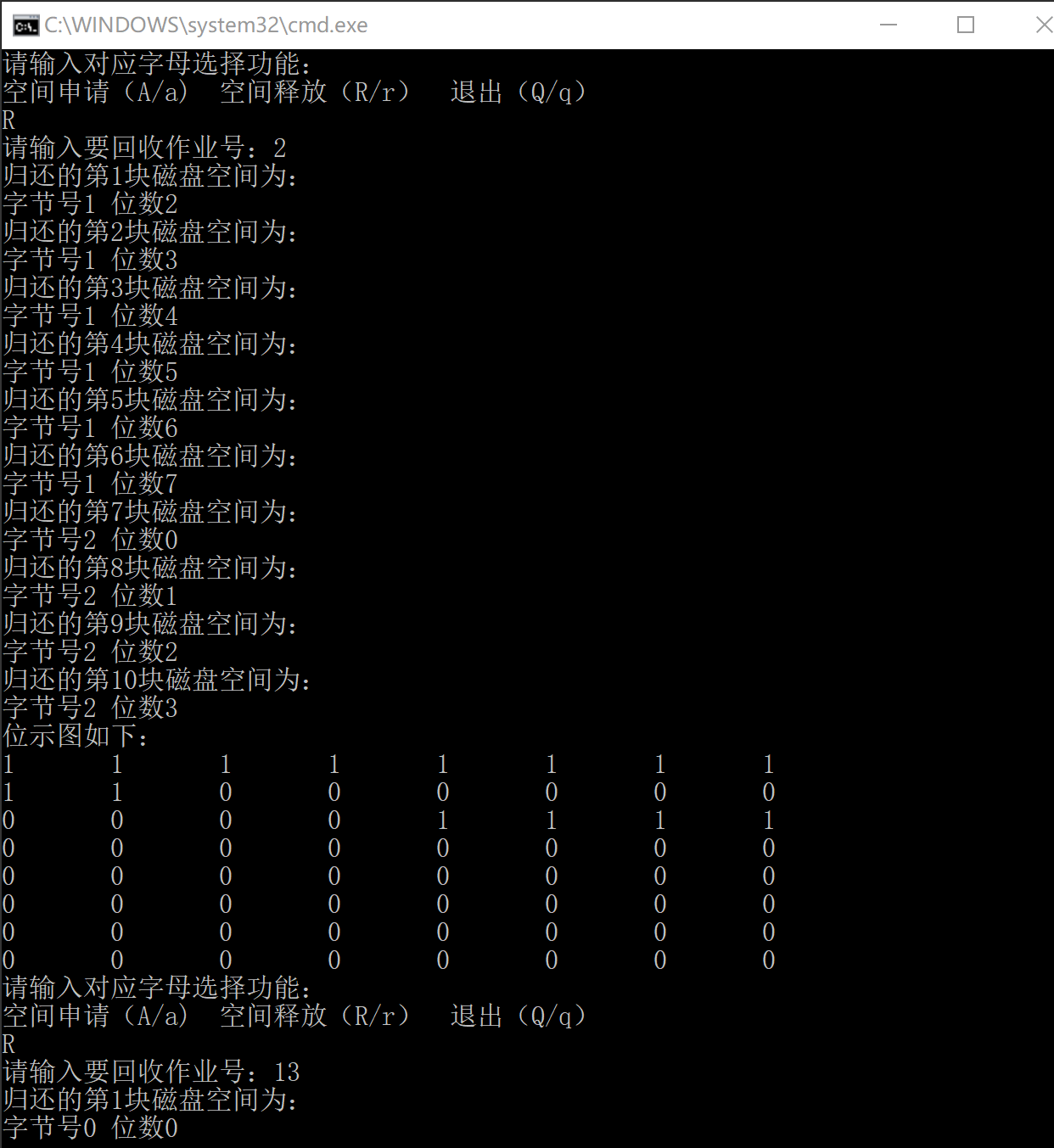


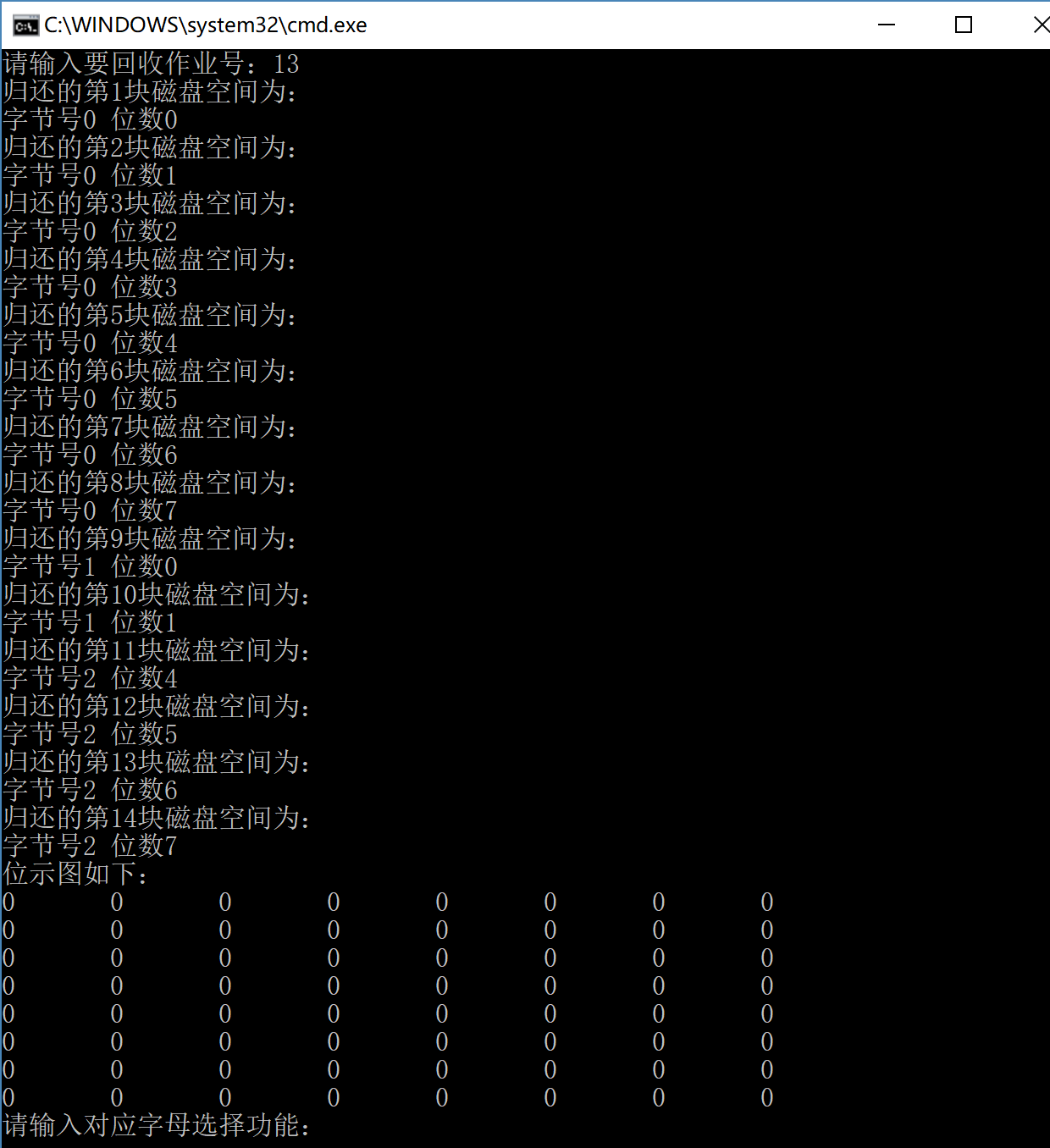


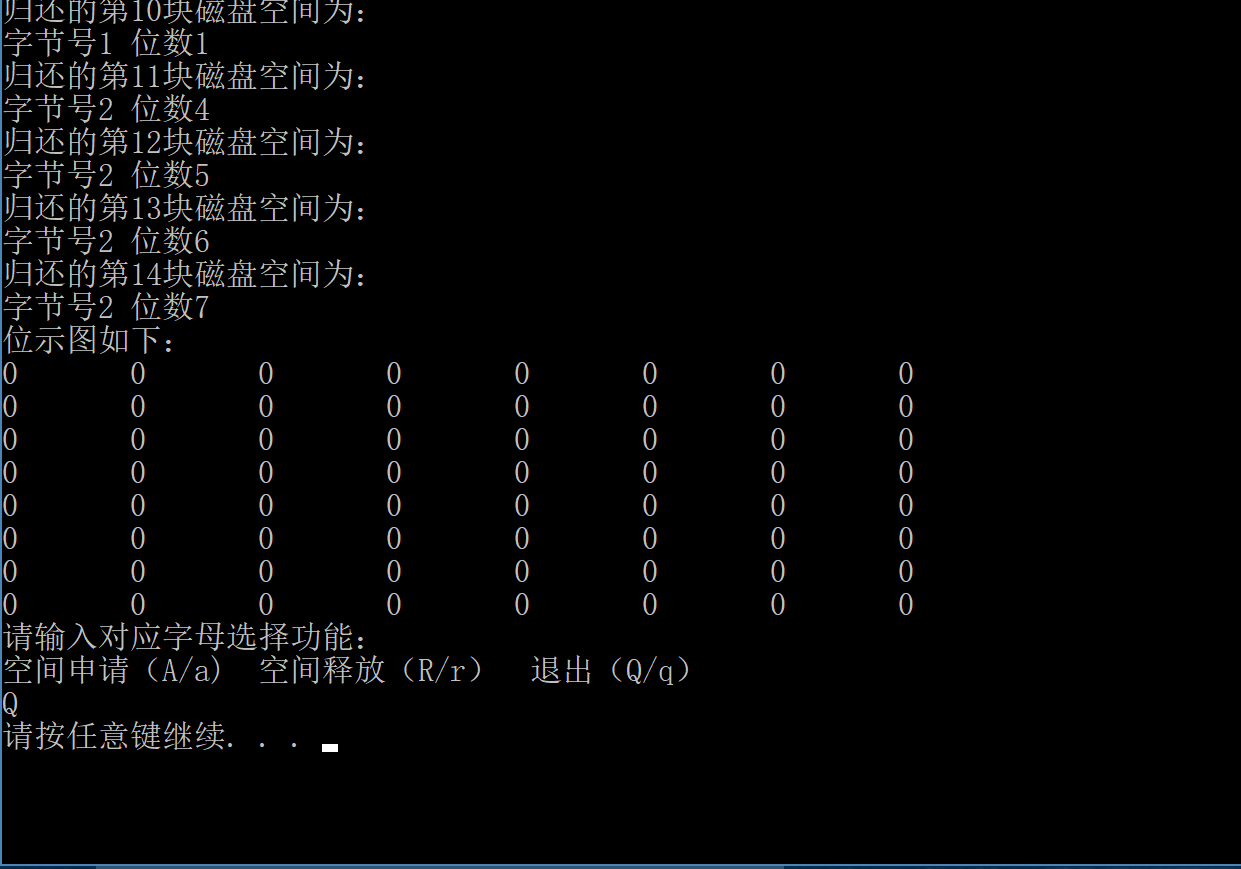












1. **总结**
2. 实习中遇到的问题及解决方法
   1. 问题：用户输入错误时由于cin的关系更容易产生问题

解决方法：输入要加上判断语句并有所提示，如选择功能时输入错误重新选择，对cin仔细分析

* 1. 问题：开始没有考虑到用户可能会添加已存在的作业号，或者删除不存在的作业号，如果不对进程序列进行遍历查找是否存在，则会出现错误；

解决方法：每次先判断是否存在，再根据情况给出对应提示

c） 问题：开始时界面不直观，提示信息不足

解决方法：增加信息显示内容，优化界面

1. 实习收获及体会

在实验2的基础上，用新方法位示图来表示，对作业的分配回收方法有了更多了解与实践。