**《程序设计基础1》实验报告**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **大一计算机类01班** | | **姓名** | | **孙莹莹** | **学号** | **20204102** |
| **实验题目** | **名字绑定、简单的分治算法** | | | | | | | |
| **实验时间** | **2020年10月25日** | | **实验地点** | | **DS1421** | | | |
| **学年学期** | **2020-2021（1）** | | **实验性质** | | **□验证性 ■设计性 □综合性** | | | |
| 一、实验目的  1． 理解并应用名字绑定的思想解决实际问题；  2． 理解并实现简单的分治算法（折半查找）。  3． 熟悉while循环的用法。 | | | | | | | | |
| 二、实验项目内容  1. 运行下述程序，会发现结果与预期不太一致：对m[0][1]的修改同时也导致的m[1][1]及m[2][1]的修改。请: a. 执行下述代码，给出运行结果；解释造成该结果的原因；添加适当的代码证明你的理论，并给出运行结果截图。（提示：可以打印id(m[0]),id[m[1]],id[m[2]]出来看看）。  m = [[0]\*4]\*3   #生成一个3行4列的矩阵，全部元素为0 m[0][1] = 99    #0行1列元素赋值99 for x in m:    print(x)  2. 折半查找：1个列表里存储了20个子列表，各子列表内存储了学生的学号及姓名两个元素，两个元素都是字符串类型。现已知该20个学生子列表已按学号递增序排好序。请设计一个程序，使用折半查找算法通过最少次数的比较找到指定学号的学生，如果没有，报告未找到。  数据示例：[ ['201801', '张三'], ['201822', 'Andy Lee'], ... ,['20189X','Austin Hu'] ]  折半查找算法：先把被查找值与序列中间位置的元素比较，如果相等表示已找到。如果查找值 < 中位元素，这说明查找值在中位元素的左边，如果查找值 > 中位元素，则说明查找值到中位元素的右边。每进行一比较，我们大概可以把查找范围缩小一半。这种同中位元素进行比较的方法可以一直持续下去，直到找到或者发现找不到为止。最坏情况下，我们要进行logN次比较。 | | | | | | | | |
| 三、实验的算法（伪代码或者流程图）和源程序  实验一：  m = [[0]\*4]\*3   #生成一个3行4列的矩阵，全部元素为0  m[0][1] = 99    #0行1列元素赋值99  for x in m:     print(x)    m = [1,2,3,4,5]  n = m  n[2] = 0  print(id(n),n)  print(id(m),m)  实验二：  classmate = [['20200001','a'],['20200002','b'],['20200003','c'],['20200004','d'],['20000005','e'],['20200006','f'],['20200007','g'],['20200008','h'],['20200009','i'],['20200010','j'],['20200011','k'],['20200012','l'],['20200013','m'],['20200014','n'],['20200015','o'],['20200016','p'],['20200017','q'],['20200018','r'],['20200019','s'],['20200020','t']]  def binarySearch(s, v):      begin = 20200001      end = 20200020      while end >= begin:          mid = (begin + end) // 2            if v == mid:              return mid          elif v > mid:              begin = mid + 1          else:              end = mid - 1      return -1  idx = binarySearch(20200000,int(input('输入你的学号：')))  if idx < 0:      print("Not Found.")  else:      print("Found: ", classmate[idx-20200001][1]) | | | | | | | | |
| 四、实验结果及分析和（或）源程序调试过程、实验总结与体会  实验一：        实验中开始没有找到思路该如何解释这种现象，根据提示输入id()后发现他们的id是一样的，这时考虑到有可能是名字绑定的相关内容，那么就需要设计一个程序证明列表在用=赋值时不会生成列表副本，而是进行了绑定。那么只需要设计一个简单列表，对其通过=赋值改变后，打印出先后两个列表的id，若id相同则可证明上述结论。  实验结论是: python将列表一个变量赋值到另外一个变量时不会复制出列表的代替品，而是将列表关联到另一个变量上。而在生成一个行列式时，如果改变第一行的某个元素，通过名字绑定及后续的操作，就是对所有行的同一列元素进行了修改。  通过这个实验，我体会到问题的解决可以化归为最初始简单的知识点，通过基本知识点去解决复杂的问题。同时找到合适的思路设计实验证明自己的结论也十分重要。列表在用=赋值时，不会生成列表的副本，而是改变原来的列表，这个原则可以运用在单行列表与生成行列式中。  实验二：    实验中遇到的问题是当解决的问题是列表嵌套时，如何从老师教的例子中灵活变换。折半查找法的主要思路就是运用元素的编号进行大小比较对半查找，那如果将元素编号换成学号，begin和end进行相应的改变，最后输出时进行加减操作输出相应列表中的元素即可。  实验结论是：对半查找法可以运用在列表及列表的嵌套中进行元素的查找。如果运用在列表中（列表的嵌套中如果时通过被嵌套列表在列表中对应的编号进行查找也可），可以运用相应元素对应在列表中的编号进行操作。若运用在列表嵌套，如本例通过列表中的数字元素进行查找时，则将相应的begin和end进行改变，最后输出时通过一定加减变换即可输出对应元素。  心得是遇到问题时可以从多方面去考虑，而不是拘泥于老师讲的例子，而是要灵活运用。 | | | | | | | | |

实验报告填写说明：

1、第一、二部分由老师提供；

2、第三部分填写源程序和算法，源程序要符合程序编写风格（缩进、注释等）；

3、第四部分主要填写程序结果（截图）、解决问题的方法、总结和体会等；

4、报告规范：包含报告页眉、报告的排版、内容是否填写，命名是否规范等。

5、源程序和实验报告命名：学号姓名序号.py 学号姓名序号.docx，例如学号20181234的张三同学，**他的第一次实验命名为：20181234张三1.py和20181234张三1.docx**

**6、向蓝墨云提交两个文件，一个实验报告，一个程序文件（代码合并）**