苏州大学实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学院 | 计算机科学与技术学院 | 专业 | 计算机科学与技术 |
| 学号 | 2110410049 | 姓名 | 鲍科臻 |
| 课程名称 | Python程序设计 | 指导老师 | 朱晓旭 |
| **实验一 数据容器综合应用1** | | | |
| **一．实验目的**   1. 掌握列表、元组、集合和字典等数据容器的使用方法。 2. 掌握通过多个步骤，完成较复杂的处理过程。   **二．实验内容**  已知一个包含了若干个整数的列表A，以及一个包含了2个整数的列表B。   1. 编写一个函数，保留列表A中包含数字3或数字7的所有素数，其余整数删除。 2. 编写一个函数，将上述第一步中所得到的列表A中所有素数按顺序以两个数构成二维平面上点的坐标，如果最后存在单个素数，则丢弃。例如：素数：563、773、379、631，577，则(563,773)构成一个坐标点，(379,631)构成一个坐标点，577因为是单个素数，则丢弃。 3. 编写一个函数，将列表B中的这个两个数构成二维平面上一个点P的坐标，比较得到第二步中所得到的所有点中和P点距离最远的点（有且仅有1个），并以列表的形式返回距离最远点结果。例如：如果(563,773)是所求得的距离最近点，那么返回结果为：[563,773]   **答题提示：**   1. 某些需要复用的功能，可以单独编写函数供其他函数调用，如：将判断素数、判断某个整数中是否包含某个数字等。   **三、其他说明**   1. 在给定的sample.py的基础上完善程序。 2. 程序主体部分不能修改，必须以Try01()函数作为整个函数的入口。   **四、实验程序（含必要的注释）及测试结果**  import math  def is\_prime(n):      if n < 2:          return False      for i in range(2, int(math.sqrt(n)) + 1):          if n % i == 0:              return False      return True  def contains\_3\_or\_7(n):      return '3' in str(n) or '7' in str(n)  def s1(A):      result = []      for num in A:          if is\_prime(num) and contains\_3\_or\_7(num):              result.append(num)      return result  def s2(A):      if len(A) % 2 == 1:          A.pop()      points = []      for i in range(0, len(A), 2):          points.append((A[i], A[i + 1]))      return points  def s3(points, B):      P = tuple(B)      max1 = -1      farthest = None      for point in points:          distance = math.sqrt((point[0] - P[0]) \*\* 2 + (point[1] - P[1]) \*\* 2)          if distance > max1:              max1 = distance              farthest = point      return list(farthest)  def Try01(A, B):      filtered\_primes = s1(A)      points = s2(filtered\_primes)      if not points:          return []      farthest\_point = s3(points, B)      return farthest\_point  if \_\_name\_\_ == '\_\_main\_\_':      print(Try01([2, 68, 4, 3, 5, 7, 90, 11, 13], [1, 1]))      print(Try01([2, 68, 4, 3, 5, 7, 90, 11, 13, 17], [1, 1]))      print(Try01([37, 2, 68, 4, 3, 5, 7, 90, 11, 13, 17, 19, 23, 29, 31, 37, 400, 37, 37], [37, 37]))      print(Try01([37, 2, 68, 4, 3, 5, 7, 90, 11, 13, 17, 19, 23, 29, 31, 37, 400, 37, 37], [1, 1]))      print(Try01([2, 68], [1, 1]))      print(Try01([2, 5, 68], [1, 1]))      print(Try01([2, 5, 7, 68], [1, 1]))      print(Try01([2, 3, 7, 68], [1, 1]))      print(Try01([2, 3, 3, 7, 68], [1, 1]))      print(Try01([2, 3, 3, 7, 9, 68], [1, 1]))    **五、实验总结**  在实验中学会了函数的调用，掌握列表、元组、集合和字典等数据容器的使用方法 | | | |