#Inläsning (flera vid svåra inläsningar)  
#Beräkning (flera vid svåra beräkningar)  
#Presentation  
  
#Categorizes CSV. Each batch[0] contains their coordinates[1:3] and measurement[3].  
def categorize\_batches(csv\_file):  
  
 batches\_sorted = dict()  
  
 with open(csv\_file, 'r') as h:  
 for line in h:  
 four\_values = line.split(',')  
 batch = four\_values[0]  
  
 if not batch in batches\_sorted:  
 batches\_sorted[batch] = []  
  
 batches\_sorted[batch] += [(float(four\_values[1]), float(four\_values[2]), float(four\_values[3]))]  
 return batches\_sorted  
  
#Calculates the average of measurements for each batch for coordinates within scope  
def calculating\_average\_measurement(sorted\_data):  
  
 average\_measurement\_batch = {}  
  
 for batch, coordinates\_measurement in sorted\_data.items():  
 if len(coordinates\_measurement) > 0:  
 acceptable\_batches = 0  
 sum\_of\_measurements = 0  
  
 for (coordinate1, coordinate2, measurement) in coordinates\_measurement:  
 if coordinate1\*\*2 + coordinate2\*\*2 <= 1:  
 sum\_of\_measurements += measurement  
 acceptable\_batches += 1  
 average\_measurement = sum\_of\_measurements/acceptable\_batches  
 average\_measurement\_batch[batch] = average\_measurement  
  
 else:  
 average\_measurement\_batch[batch] = None  
 return average\_measurement\_batch  
  
#Presents the batch results  
def batch\_result(batch\_result):  
 for batch, average in batch\_result.items():  
 print(batch, "\t", average)  
  
  
def main(): #Asks for input  
 data\_file = input('Which data file? ')  
 categorized = categorize\_batches(data\_file)  
 average\_batch = calculating\_average\_measurement(categorized)  
 batch\_result(average\_batch)  
  
  
  
#Starts program  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()