**import** codigosLivros  
  
**def** traduzListaLivros(listaLivros):  
 texto = **""  
  
 for** item **in** listaLivros:  
 texto+=codigosLivros.getNomeLivro(item)+**","  
 return** texto  
  
  
**def** traduzRegra(antecedente,consequente):  
 textoAntecedente = **""  
  
 for** item **in** antecedente:  
 textoAntecedente+=codigosLivros.getNomeLivro(item)+**","** textoAntecedente += **" --> "  
  
 for** item **in** consequente:  
 textoAntecedente+=codigosLivros.getNomeLivro(item)+**","  
 return** textoAntecedente  
  
  
  
**def** leituraListaLivrosUsuario():  
 print(**'Por favor insira uma lista com os livros que vc mais gostou de acordo com a lista abaixo:'**)  
 i = 1  
 **for** livro **in** codigosLivros.livroCodigo:  
 print(**'%s:%s '** %(i ,codigosLivros.getNomeLivro(livro)))  
 i+=1  
  
 listaEntrada = input()  
 entrada = listaEntrada.split(**','**)  
 linha = []  
  
 **for** livroGostado **in** entrada:  
 linha.append(str(**"Lv"**+livroGostado))  
  
 texto = traduzListaLivros(linha)  
 print(texto)  
 **return** linha  
  
  
**def** carregaRegras():  
 **with** open(**'regrasEncontradas.csv'**,**'r'**)**as** arq:  
 conjuntoDeregras = arq.readlines()  
 **return** conjuntoDeregras  
  
  
**def** geraRecomendacao(antecedente,consequente,confianca):  
 txtRecomendacao = **" É recomendado dizer (com %s %% de confiança) que quem leu "** % (str(round(float(confianca) \* 100,2)))  
  
 **for** item **in** antecedente:  
 itemNome = codigosLivros.getNomeLivro(str(item))+**", "** txtRecomendacao +=itemNome  
 txtRecomendacao += **"Também gostou de ler "  
  
 for** item **in** consequente:  
  
 txtRecomendacao += codigosLivros.getNomeLivro(str(item))  
  
  
 **return** txtRecomendacao  
  
  
  
 **return ""  
def** buscaRegrasCompativeis(livrosLidos):  
 conjuntoDeregras = carregaRegras()  
 setLivroslidos = set(livrosLidos)  
 recomendacoes = **""  
  
 for** linha **in** conjuntoDeregras:  
 regraConfianca = linha.split(**':'**)  
 regraCompleta = regraConfianca[0]  
 confianca = regraConfianca[1]  
  
 elementosRegra = regraCompleta.split(**'-->'**)  
 antecedente = elementosRegra[0].split(**','**)  
 consequente = elementosRegra[1].split(**','**)  
  
 setAntecedente = set(antecedente)  
  
  
 **if** setLivroslidos == setAntecedente:  
 recomendacoes = geraRecomendacao(antecedente,consequente,confianca)  
 print(recomendacoes)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 livrosLidos = leituraListaLivrosUsuario()  
 regrasQueContemOsLivrosLidos = buscaRegrasCompativeis(livrosLidos)