Curso:

**UPskill – Programação em Python**

Unidade de Formação:

**Programação**

Trabalhos SA

**[['2023-12-05', '14:00', '17:30', 'SA']['2023-12-12', '14:00', '17:30', 'SA']['2023-12-19', '14:00', '17:30', 'SA']**

Henrique José Alves Gonzaga

Entrega:

**2023-12-31 23:59**

Formato de entrega:

**PDF, baseado no formato existente no moodle (formato trabalho 1)**

Índice

[1 Entrega 1](#_Toc153995004)

[2 Enunciado 1](#_Toc153995005)

[3 Python 2](#_Toc153995006)

[3.1 Programas 2](#_Toc153995007)

[3.2 Resultado de execução do programa 9](#_Toc153995008)

# Entrega

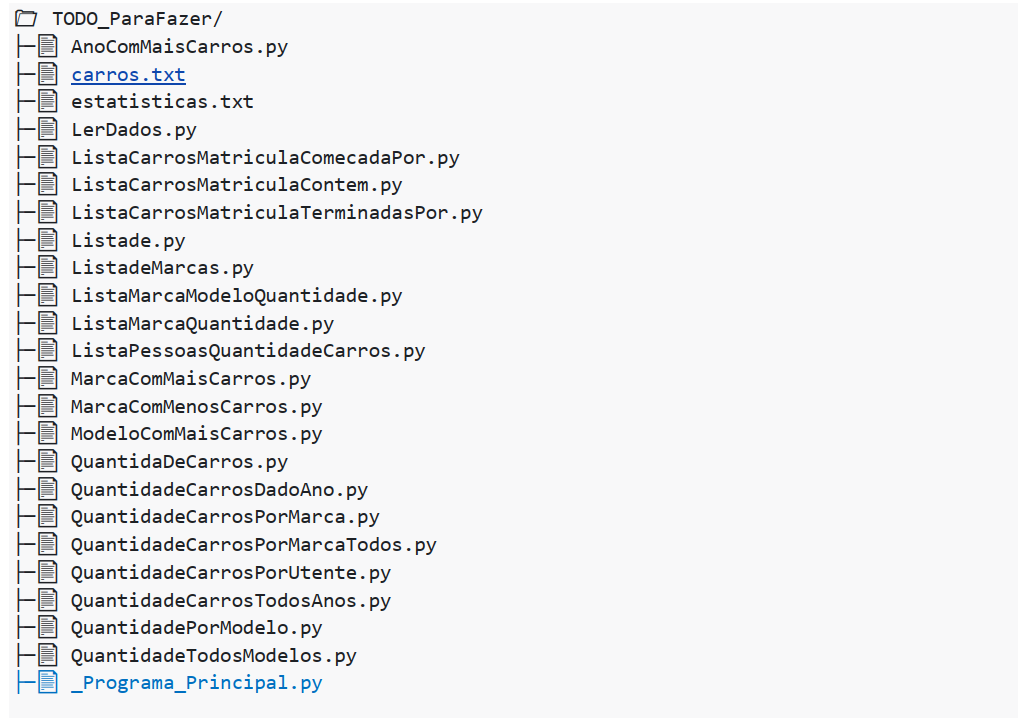
Entregar ficheiro zip com:

Nome: <Número>\_<PrimeiroNome>\_<UltimoNome>\_Carros\_Estatisticas.zip

Conteúdo: dados, programa, ficheiro de texto de resultados e relatório word/pdf

# Enunciado

Considere o ficheiro de texto CSV carros.txt e a lista de ficheiros:



Escreva as funções propostas no ficheiro Programa\_Carros.py, listadas acima. Nos ficheiros pode incluir outras funções relacionadas. Pode ainda, adicionar outras funções que respondam a questões de interesse.

No ficheiro Programa\_Principal.py, importe todas funções e escreva código para as testar. Exemplo para duas funções:

import QuantidadeCarrosPorUtente from LerDados import \* import QuantidaDeCarros f = open("estatisticas.txt", "wt", encoding='UTF-8')

*# Quantos carros existem?*

carros = LerDados("carros.txt")

qt = QuantidaDeCarros.QuantidaDeCarros(carros)

s = f"Quantidade de carros: {qt}"

print(s)

print(s, file=f)

*#-------------------------*

from QuantidadeCarrosPorUtente import \*

lista = QuantidadeCarrosPorUtente(carros)

print(len(lista), lista[0:10])

lista = UtentesComMaisCarros(carros)

print(len(lista), lista[0:10])

*#------------------------*

from ListaCarrosMatriculaComecadaPor import \*

|  |  |
| --- | --- |
| lista = ListaCarrosMatriculaComecadaPor(carros,  print(len(lista), lista[0:10]) | '99') |

f.close

# Python

## Programas

#main.py

from lerdados import \*  
from anocommaiscarros import \*  
from listacarrosmatriculacomecadapor import \*  
from listacarrosmatriculacontem import \*  
from listacarrosmatriculaterminadaspor import \*  
from listamarcamodeloquantidade import \*  
from listamarcaquantidade import \*  
from listamarcas import \*  
from listamodelo import \*  
from marcacommaiscarros import \*  
from marcacommenoscarros import \*  
from modelocommaiscarros import \*  
from quantidadecarros import \*  
from quantidadecarrosdadoano import \*  
from quantidadecarrosmarca import \*  
from quantidadecarrosmodelo import \*  
from quantidadecarrospormarcatodos import \*  
from quantidadecarrosporutente import \*  
from quantidadecarrostodosanos import \*  
from quantidadeordenadatodosmodelos import \*  
  
  
ano\_mais\_carros = anocommaiscarros(carros)  
print(f"Ano: {ano\_mais\_carros[0]} - quantidade: {ano\_mais\_carros[1]}")  
lista\_ano\_carros = listaano(carros)  
print(lista\_ano\_carros)  
matricula\_comeca = listacarrosmatriculacomecadapor(carros, '88')  
print(matricula\_comeca[:10])  
matricula\_contem = listacarrosmatriculacontem(carros, 'fg')  
print(matricula\_contem[:10])  
matricula\_termina = listacarrosmatriculaterminadaspor(carros, '85')  
print(matricula\_termina[:10])  
lista\_resultado = listamarcamodeloquantidade(carros)  
print(lista\_resultado)  
lista\_marca\_quantidade = listamarcaquantidade(carros)  
print(lista\_marca\_quantidade)  
lista\_marcas = listamarcas(carros)  
print(lista\_marcas)  
lista\_modelo = listamodelo(carros)  
print(lista\_modelo)  
marca\_mais\_carros = marcacommaiscarros(carros)  
print(f"Marca c/ mais carros: {marca\_mais\_carros[0]}; quantidade: {marca\_mais\_carros[1]}")  
marca\_menos\_carros = marcacommenoscarros(carros)  
print(f"Marca c/ menos carros: {marca\_menos\_carros[0]}; quantidade: {marca\_menos\_carros[1]}")  
modelo\_mais\_carros = modelocommaiscarros(carros)  
print(f"Modelo c/ mais carros: {modelo\_mais\_carros[0]}; quantidade: {modelo\_mais\_carros[1]}")  
print(f"Quantidade de carros registrados: {quantidadecarros(carros)}")  
quantidade\_carros\_ano = quantidadecarrosdadoano(carros, '2007')  
print(f"ano a verificar: {quantidade\_carros\_ano[0]}; quantidade: {quantidade\_carros\_ano[1]}")  
marca\_verificar = 'BMW'  
qtde\_carros\_marca = quantidadecarrosmarca(carros, marca\_verificar)  
print(f"Marca a verificar: {marca\_verificar}; quantidade: {qtde\_carros\_marca}")  
modelo = 'corsa'  
quantidade\_carros\_modelo = quantidadecarrosmodelo(carros, modelo)  
print(f"Modelo a verificar: {modelo}; quantidade: {quantidade\_carros\_modelo}")  
quantidade\_carros\_por\_marca = quantidadecarrospormarcatodos(carros)  
print(quantidade\_carros\_por\_marca)  
qtde\_carros\_utente = quantidadecarrosporutente(carros)  
print(qtde\_carros\_utente[:10])  
carros\_por\_ano = quantidadecarrostodosanos(carros)  
print(carros\_por\_ano)  
qtde\_ordenada\_modelos = quantidadeordenadatodosmodelos(carros)  
print(qtde\_ordenada\_modelos)

# anocommaiscarros.py

from listaano import \*  
from quantidadecarrosano import \*  
  
  
# Ano com mais carros  
def anocommaiscarros(carros):  
 lista = []  
 carro\_ano = listaano(carros)  
 maior\_numero = 0  
 ano\_mais\_carros = ''  
 for ano in carro\_ano:  
 quantidade = quantidadecarrosano(carros, ano)  
 lista.append([ano, quantidade])  
 for i, v in enumerate(lista):  
 if v[1] > maior\_numero:  
 maior\_numero = v[1]  
 ano\_mais\_carros = v[0]  
 return ano\_mais\_carros, maior\_numero

# lerdados.py

def lerdados(nome\_arquivo):  
 lista = []  
 with open(nome\_arquivo, 'rt', newline='', encoding='utf -8') as f:  
 reader = csv.reader(f, delimiter=';')  
 reader.\_\_next\_\_() # ignora o cabeçalho, salta a linha  
 for r in reader:  
 lista.append(r)  
 return lista  
  
  
carros = lerdados('carros.txt')

# listaano.py

def listaano(lista):  
 lista\_ano = []  
 for i in lista:  
 verifica\_ano = i[3]  
 if verifica\_ano not in lista\_ano:  
 lista\_ano.append(verifica\_ano)  
 return sorted(lista\_ano, reverse=True)

# listacarrosmatriculacomecadapor.py

def listacarrosmatriculacomecadapor(carros, texto):  
 lista = []  
 for i in carros:  
 verifica\_matricula = i[0]  
 if texto == verifica\_matricula[:2]:  
 lista.append(i)  
 return lista

# listacarrosmatriculacontem.py

def listacarrosmatriculacontem(carros, texto):  
 lista = list()  
 for i in carros:  
 verifica\_matricula = i[0]  
 if texto.upper() in verifica\_matricula:  
 lista.append(i)  
 return lista

# listacarrosmatriculaterminadaspor.py

def listacarrosmatriculaterminadaspor(carros, texto):  
 lista = list()  
 for i in carros:  
 verifica\_matricula = i[0]  
 if texto in verifica\_matricula[-2:]:  
 lista.append(i)  
 return lista

# listade.py

def listade(lista, coluna):  
 r = []  
 for x in lista:  
 if x[coluna] not in r:  
 r.append(x[coluna])  
 return r

# listamarcamodeloquantidade.py

def listamarcamodeloquantidade(carros):  
 lista = []  
 lista\_modelo = []  
 total\_quantidade = []  
 total\_quantidade\_modelo = []  
 for i in carros:  
 marca = i[1]  
 if marca in lista:  
 posicao = lista.index(marca)  
 total\_quantidade[posicao] += 1  
 else:  
 lista.append(marca)  
 total\_quantidade.append(1)  
 for j in carros:  
 modelo = j[2]  
 if modelo in lista\_modelo:  
 posicao\_modelo = lista\_modelo.index(modelo)  
 total\_quantidade\_modelo[posicao\_modelo] += 1  
 else:  
 lista\_modelo.append(modelo)  
 total\_quantidade\_modelo.append(1)  
 return list(zip(lista, total\_quantidade)), list(zip(lista\_modelo, total\_quantidade\_modelo))

# listamarcaquantidade.py

from listamarcas import \*  
from quantidadecarrosmarca import \*  
  
  
def listamarcaquantidade(carros):  
 lista = []  
 marca = listamarcas(carros)  
 for i in marca:  
 quantidade = quantidadecarrosmarca(carros, i)  
 lista.append([quantidade, i])  
 sorted\_lista = sorted(lista, reverse=True)  
 return sorted\_lista

# listamarcas.py

def listamarcas(lista):  
 lista\_marcas = []  
 for i in lista:  
 verifica\_marca = i[1]  
 if verifica\_marca not in lista\_marcas:  
 lista\_marcas.append(verifica\_marca)  
 return lista\_marcas

# listamodelo.py

def listamodelo(lista):  
 lista\_modelo = []  
 for i in lista:  
 verifica\_modelo = i[2]  
 if verifica\_modelo not in lista\_modelo:  
 lista\_modelo.append(verifica\_modelo)  
 return lista\_modelo

# marcacommaiscarros.py

from listamarcas import \*  
from quantidadecarrosmarca import \*  
  
  
# Marca com mais carros  
def marcacommaiscarros(carros):  
 lista = []  
 marcas = listamarcas(carros)  
 maior\_valor = 0  
 marca\_maior\_numero = ''  
 for marca in marcas:  
 quantidade = quantidadecarrosmarca(carros, marca)  
 lista.append([marca, quantidade])  
 for i, v in enumerate(lista):  
 if v[1] > maior\_valor:  
 maior\_valor = v[1]  
 marca\_maior\_numero = v[0]  
 return marca\_maior\_numero, maior\_valor

# marcacommenoscarros.py

from listamarcas import \*  
from quantidadecarrosmarca import \*  
  
  
def marcacommenoscarros(carros):  
 lista = []  
 marcas = listamarcas(carros)  
 menor\_numero = 9999999  
 marca\_menor\_numero = ''  
 for marca in marcas:  
 quantidade = quantidadecarrosmarca(carros, marca)  
 lista.append([marca, quantidade])  
 for i, v in enumerate(lista):  
 if v[1] < menor\_numero:  
 menor\_numero = v[1]  
 marca\_menor\_numero = v[0]  
 return marca\_menor\_numero, menor\_numero

# modelocommaiscarros.py

from listamodelo import \*  
from quantidadecarrosmodelo import \*  
  
  
def modelocommaiscarros(carros):  
 lista = []  
 modelo = listamodelo(carros)  
 maior\_numero = 0  
 modelo\_maior\_numero = ''  
 for i in modelo:  
 quantidade = quantidadecarrosmodelo(carros, i)  
 lista.append([i, quantidade])  
 for i, v in enumerate(lista):  
 if v[1] > maior\_numero:  
 maior\_numero = v[1]  
 modelo\_maior\_numero = v[0]  
 return modelo\_maior\_numero, maior\_numero

# quantidadecarros.py

# Quantos carros existem?  
def quantidadecarros(lista):  
 return len(lista)

# quantidadecarrosano.py

def quantidadecarrosano(lista, ano\_contar):  
 total\_quantidade = 0  
 for i in lista:  
 ano = i[3]  
 if ano == ano\_contar:  
 total\_quantidade += 1  
 return total\_quantidade

# quantidadecarrosdadoano.py

# carros por um dado ano  
def quantidadecarrosdadoano(carros, ano):  
 quantidade\_total = 0  
 for i in carros:  
 verifica\_ano = i[3]  
 if verifica\_ano == ano:  
 quantidade\_total += 1  
 return ano, quantidade\_total

# quantidadecarrosmarca.py

def quantidadecarrosmarca(lista, marca\_contar):  
 total\_quantidade = 0  
 for i in lista:  
 marca = i[1]  
 if marca == marca\_contar:  
 total\_quantidade += 1  
 return total\_quantidade

# quantidadecarrosmodelo.py

def quantidadecarrosmodelo(lista, modelo\_contar):  
 total\_quantidade = 0  
 for i in lista:  
 modelo = i[2]  
 if modelo.lower() == modelo\_contar:  
 total\_quantidade += 1  
 return total\_quantidade

# quantidadecarrospormarcatodos

from listamarcas import \*  
from quantidadecarrosmarca import \*  
  
  
def quantidadecarrospormarcatodos(lista):  
 r = []  
 marcas = listamarcas(lista)  
 for m in marcas:  
 qt = quantidadecarrosmarca(lista, m)  
 r.append([m, qt])  
 return sorted(r)

# quantidadecarrosporutente.py

# Quantidade de carros por pessoa (IDUtente)  
def quantidadecarrosporutente(carros):  
 lista = []  
 total\_quantidade = []  
 for i in carros:  
 idutente = i[4]  
 if idutente in lista:  
 posicao = lista.index(idutente)  
 total\_quantidade[posicao] += 1  
 else:  
 lista.append(idutente)  
 total\_quantidade.append(1)  
 return sorted(list(zip(lista, total\_quantidade)), reverse=True)  
  
  
def utentecommaiscarros(carros):  
 return quantidadecarrosporutente(carros)[0][1]  
  
  
def utentecommaiscarrosquantidade(carros):  
 return (quantidadecarrosporutente(carros)[0][1],  
 quantidadecarrosporutente(carros)[0][0])  
  
  
def utentescommaiscarros(carros):  
 lista = []  
 # devolver todos os elementos do vetor cujo nº carros  
 # = utente com mais carros  
 qt\_mais = utentecommaiscarrosquantidade(carros)[1] - 1  
 c = 0  
 for x in quantidadecarrosporutente(carros)[1:]: # qt, utente  
 if x[0] == qt\_mais:  
 lista.append(x[1])  
 elif x[0] < qt\_mais:  
 break  
 c = c + 1  
 print(c)  
 return lista  
  
  
def utentestopncommaiscarros(carros, top):  
 return utentescommaiscarros(carros)[:top]

# quantidadecarrostodosanos.py

from listaano import \*  
from quantidadecarrosano import \*  
  
# lista carros por ano  
def quantidadecarrostodosanos(carros):  
 lista = []  
 ano = listaano(carros)  
 for i in ano:  
 quantidade = quantidadecarrosano(carros, i)  
 lista.append([i, quantidade])  
 return lista

# quantidadeordenadatodosmodelos.py

from listamodelo import \*  
  
  
def quantidadeordenadatodosmodelos(carros):  
 lista = []  
 modelo = listamodelo(carros)  
 quantidade = 0  
 for i in modelo:  
 if i in modelo:  
 quantidade += 1  
 lista.append([quantidade, i])  
 sorted\_lista = sorted(lista, key=lambda x: x[1])  
 return sorted\_lista

## Resultado de execução do programa

C:\Users\henri\AppData\Local\Microsoft\WindowsApps\python3.10.exe C:/Users/henri/OneDrive/UpSkill/Programacao/carros/main.py

Ano: 2007 - quantidade: 523

['2010', '2009', '2008', '2007', '2006', '2005', '2004', '2003', '2002', '2001', '2000', '1999', '1998', '1997', '1996', '1995', '1994', '1993', '1992', '1991', '1990']

[['88-KV-48', 'Mitsubishi', 'Lancer', '2000', '5906074', '73'], ['88-NE-57', 'Renault', 'Espace', '2002', '8164861', '103'], ['88-VK-55', 'Peugeot', '205', '2005', '11932088', '187'], ['88-AD-09', 'Seat', 'Cordoba', '1992', '6249536', '477'], ['88-RI-36', 'Fiat', 'Punto', '2008', '8900725', '483'], ['88-ED-28', 'Audi', 'A2', '2000', '5573626', '551'], ['88-JJ-18', 'Fiat', '600', '2006', '12948841', '563'], ['88-EQ-82', 'Renault', 'Espace', '1999', '5239249', '653'], ['88-TL-92', 'Opel', 'Corsa', '1991', '8789801', '761'], ['88-MG-81', 'Opel', 'Kadet', '1997', '7315561', '764']]

[['65-FG-41', 'Opel', 'Corsa', '2007', '4562034', '210'], ['27-FG-26', 'Citroen', 'C4', '2010', '2961974', '391'], ['59-FG-26', 'Audi', 'A4', '1990', '13418698', '618'], ['57-FG-55', 'Fiat', '600', '1999', '7539001', '1773'], ['89-FG-00', 'Opel', 'Kadet', '1990', '12616189', '3119'], ['48-FG-06', 'BMW', '320', '1995', '4768401', '3875'], ['98-FG-69', 'Audi', 'A5', '2008', '6272518', '5273'], ['81-FG-02', 'Porche', 'Carrera', '2001', '3552465', '5521'], ['41-FG-53', 'Ferrari', 'Testarrosa', '2001', '3320430', '5656'], ['48-FG-84', 'Seat', 'Ibiza', '2009', '9536886', '6646']]

[['80-TM-85', 'Mitsubishi', 'Colt', '2006', '11750312', '33'], ['94-LI-85', 'Audi', 'A8', '2006', '6514918', '49'], ['87-DB-85', 'Audi', 'A2', '1999', '1789360', '96'], ['74-ES-85', 'Seat', 'Marbella', '1990', '6644496', '116'], ['24-SM-85', 'Citroen', 'C5', '1999', '9733248', '130'], ['84-JM-85', 'Peugeot', '205', '1997', '14124976', '278'], ['01-OU-85', 'Fiat', 'Panda', '2003', '11759590', '284'], ['85-RL-85', 'Citroen', 'C5', '2002', '12464572', '378'], ['09-NP-85', 'Bentley', 'Continental', '1990', '5989180', '613'], ['52-NF-85', 'Audi', 'A1', '2005', '14766748', '669']]

([('Renault', 786), ('Mitsubishi', 454), ('Citroen', 1320), ('Audi', 2069), ('Porche', 574), ('Peugeot', 542), ('Opel', 538), ('Seat', 817), ('Fiat', 791), ('BMW', 1322), ('Toyota', 270), ('Ferrari', 261), ('Bentley', 256)], [('Espace', 259), ('Colt', 235), ('C4', 253), ('C2', 289), ('TT', 262), ('A8', 264), ('Cheyenne', 292), ('305', 273), ('Corsa', 257), ('A2', 241), ('Kadet', 281), ('A6', 272), ('Ibiza', 292), ('A1', 271), ('Marbella', 265), ('A5', 261), ('Panda', 264), ('A3', 246), ('600', 269), ('Kangoo', 265), ('X3', 262), ('Megane', 262), ('530', 300), ('Lancer', 219), ('320', 252), ('Carrera', 282), ('Punto', 258), ('Avensis', 270), ('Cordoba', 260), ('205', 269), ('C1', 266), ('C5', 260), ('Testarrosa', 261), ('A4', 252), ('520', 276), ('X6', 232), ('C3', 252), ('Continental', 256)])

[[2069, 'Audi'], [1322, 'BMW'], [1320, 'Citroen'], [817, 'Seat'], [791, 'Fiat'], [786, 'Renault'], [574, 'Porche'], [542, 'Peugeot'], [538, 'Opel'], [454, 'Mitsubishi'], [270, 'Toyota'], [261, 'Ferrari'], [256, 'Bentley']]

['Renault', 'Mitsubishi', 'Citroen', 'Audi', 'Porche', 'Peugeot', 'Opel', 'Seat', 'Fiat', 'BMW', 'Toyota', 'Ferrari', 'Bentley']

['Espace', 'Colt', 'C4', 'C2', 'TT', 'A8', 'Cheyenne', '305', 'Corsa', 'A2', 'Kadet', 'A6', 'Ibiza', 'A1', 'Marbella', 'A5', 'Panda', 'A3', '600', 'Kangoo', 'X3', 'Megane', '530', 'Lancer', '320', 'Carrera', 'Punto', 'Avensis', 'Cordoba', '205', 'C1', 'C5', 'Testarrosa', 'A4', '520', 'X6', 'C3', 'Continental']

Marca c/ mais carros: Audi; quantidade: 2069

Marca c/ menos carros: Bentley; quantidade: 256

Modelo c/ mais carros: 530; quantidade: 300

Quantidade de carros registrados: 10000

ano a verificar: 2007; quantidade: 523

Marca a verificar: BMW; quantidade: 1322

Modelo a verificar: corsa; quantidade: 257

[['Audi', 2069], ['BMW', 1322], ['Bentley', 256], ['Citroen', 1320], ['Ferrari', 261], ['Fiat', 791], ['Mitsubishi', 454], ['Opel', 538], ['Peugeot', 542], ['Porche', 574], ['Renault', 786], ['Seat', 817], ['Toyota', 270]]

[('9999809', 2), ('9999005', 4), ('9995007', 2), ('9994767', 2), ('9993306', 1), ('9991747', 1), ('9990674', 2), ('9989046', 5), ('9985179', 2), ('9982442', 1)]

[['2010', 488], ['2009', 473], ['2008', 419], ['2007', 523], ['2006', 468], ['2005', 494], ['2004', 473], ['2003', 479], ['2002', 443], ['2001', 499], ['2000', 483], ['1999', 506], ['1998', 450], ['1997', 464], ['1996', 481], ['1995', 494], ['1994', 459], ['1993', 482], ['1992', 457], ['1991', 489], ['1990', 476]]

[[30, '205'], [8, '305'], [25, '320'], [35, '520'], [23, '530'], [19, '600'], [14, 'A1'], [10, 'A2'], [18, 'A3'], [34, 'A4'], [16, 'A5'], [12, 'A6'], [6, 'A8'], [28, 'Avensis'], [31, 'C1'], [4, 'C2'], [37, 'C3'], [3, 'C4'], [32, 'C5'], [26, 'Carrera'], [7, 'Cheyenne'], [2, 'Colt'], [38, 'Continental'], [29, 'Cordoba'], [9, 'Corsa'], [1, 'Espace'], [13, 'Ibiza'], [11, 'Kadet'], [20, 'Kangoo'], [24, 'Lancer'], [15, 'Marbella'], [22, 'Megane'], [17, 'Panda'], [27, 'Punto'], [5, 'TT'], [33, 'Testarrosa'], [21, 'X3'], [36, 'X6']]

Process finished with exit code 0