

**Université des Sciences et de la Technologie Houari Boumédiène**

**Faculté d’Informatique**

**Département d’Intelligence Artificielle et Sciences de Données**

**Master Systèmes Informatiques intelligents**

**Module : Base de données avancées**

**RAPPORT DU Projet 02 NoSQL-MongoDB**

**Réalisé par :**

Zait Fouad 181831072145

**Partie I : Importation de la BD**

**Création de la base de données et une collection world:**

-Pour créer la base de données BDD on utilise la commande **use BDD.**

-Pour créer la collection World on utilise la commande **db.createCollection(“World”).**

**Résultat:**



**Importer les données dans cette collection :**

Pour importer les données on exécute la commande :

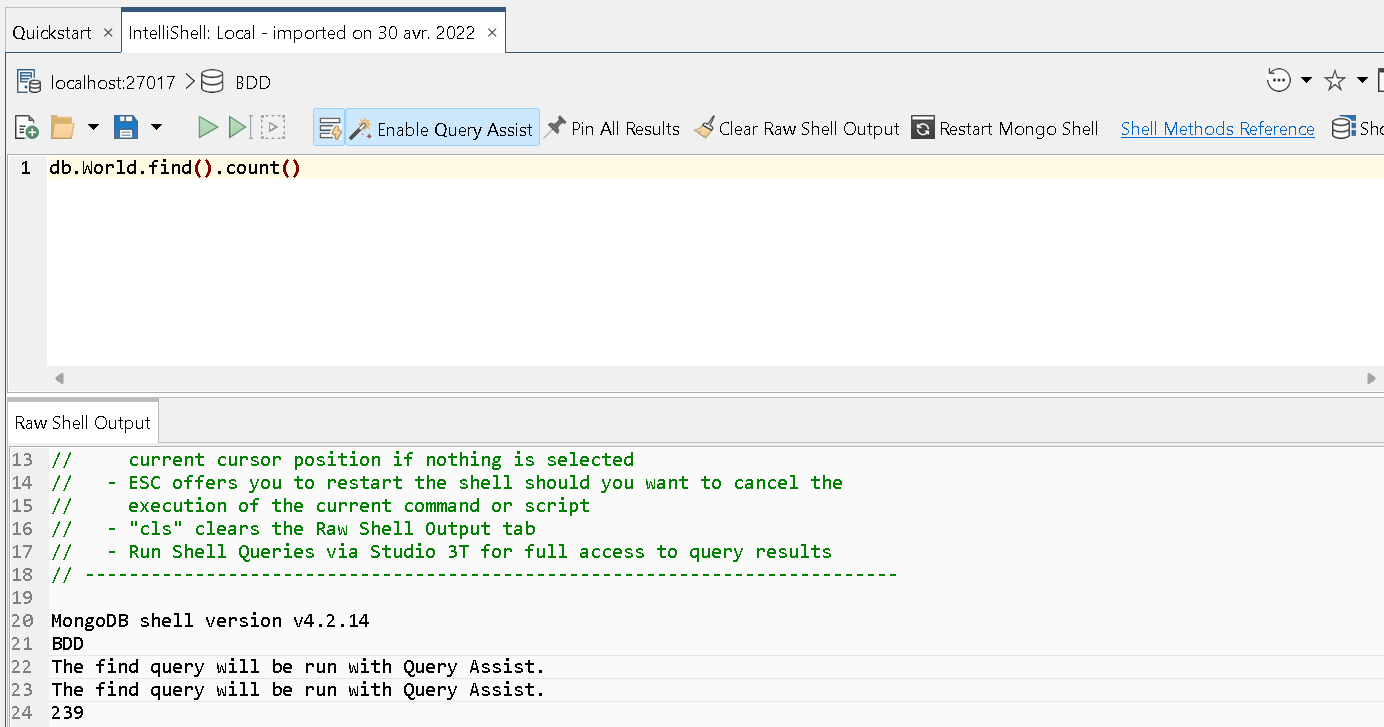
**mongoimport –jsonArray –db BDD –collection World –file “emplacement du fichier json ”**

**Résultat:**



**Vérifier que tout est ok en lançant robot3t.exe :**

**Résultat:**



À l’exécution de db.World.find().count() on trouve 239 documents donc tout est ok .

**Partie II : MongoDB – PyMongo – Python**

Connecter mongopy au serveur de la base de données :

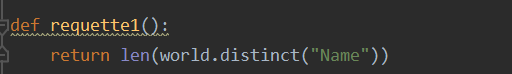
myclient = MongoClient()  
myclient = MongoClient('localhost', 27017)  
database = myclient['BDD']  
world = database['World']

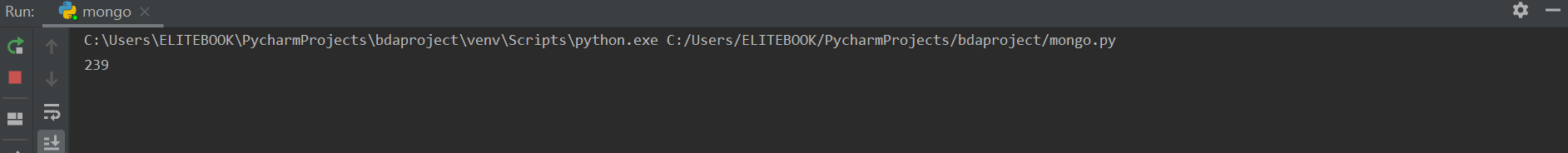
**1. Déterminer le nombre exact de pays :**

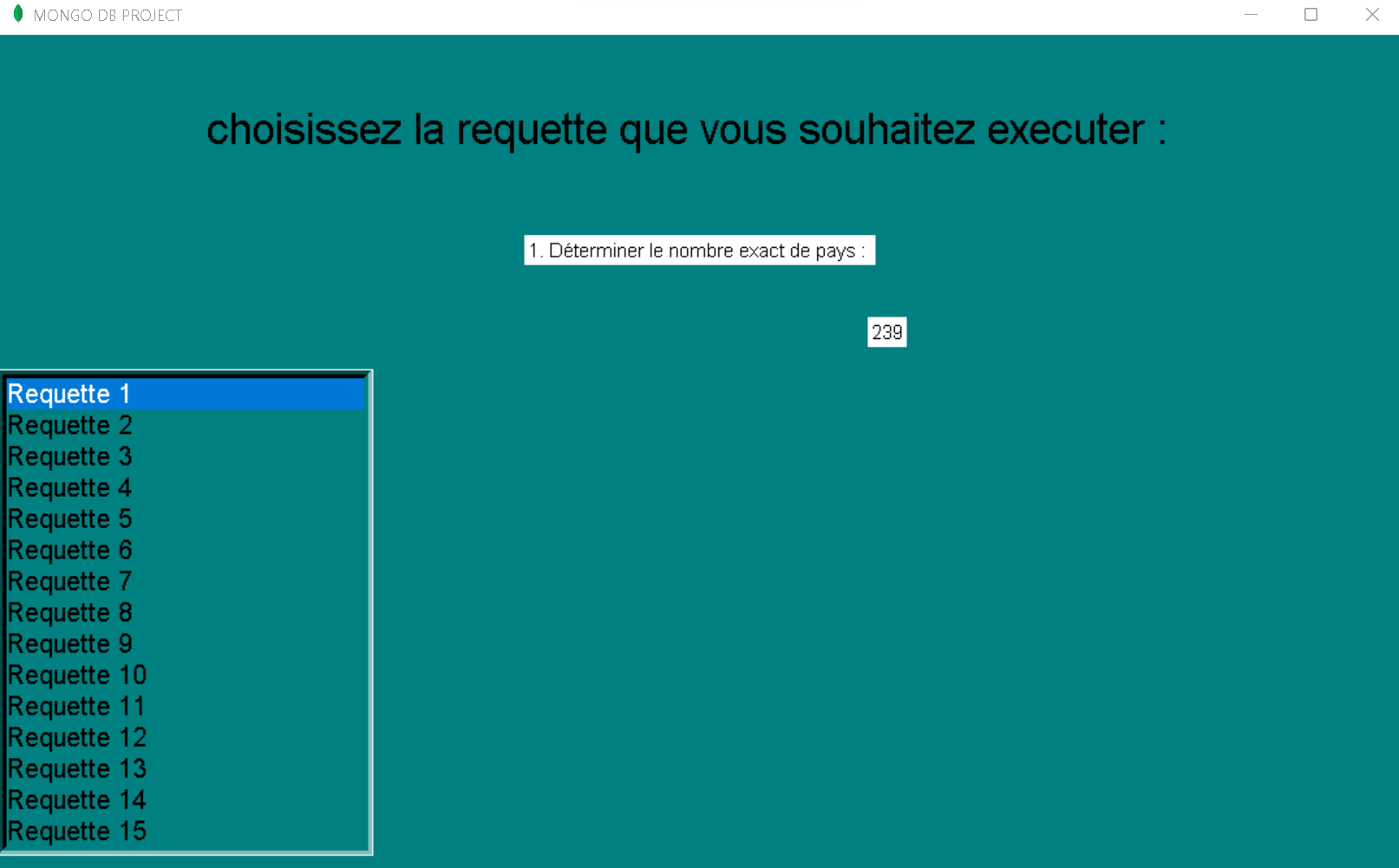
**Fonction:**

def requette1():  
 return len(world.distinct("Name"))

**Code:**



**Résultat sur le terminal:****Résultat sur l’interface :**



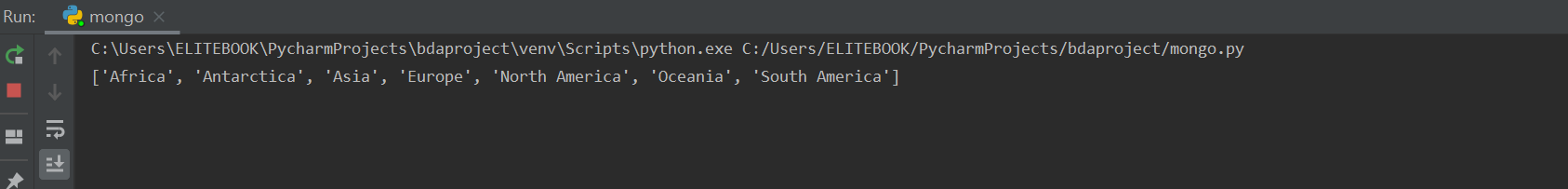
**2-Lister les différents continents :**

**Fonction:**

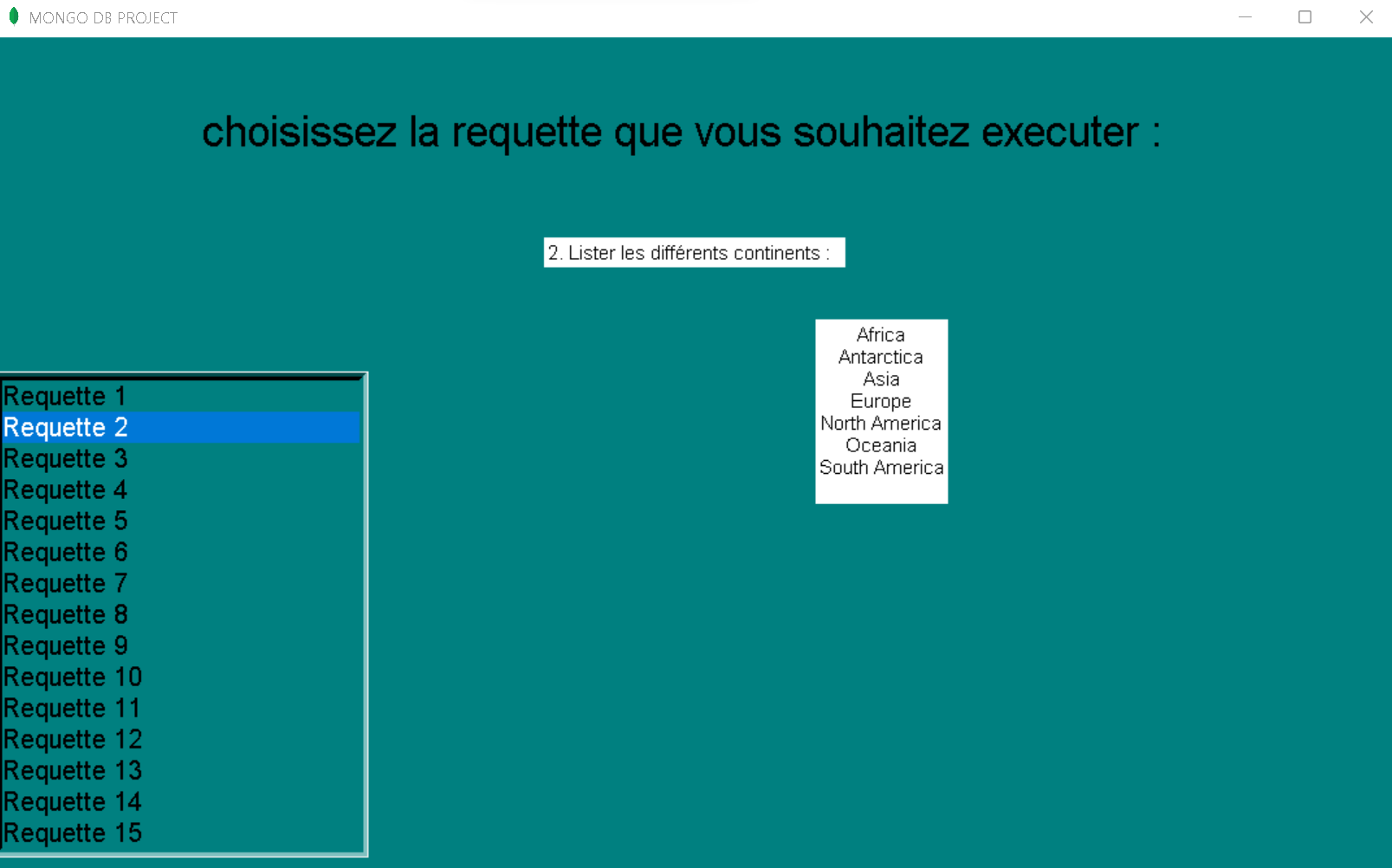
def requette2():  
 return world.distinct("Continent")

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**

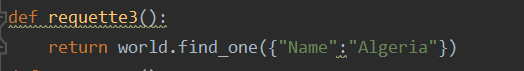


**3- Lister les informations de l’Algérie :**

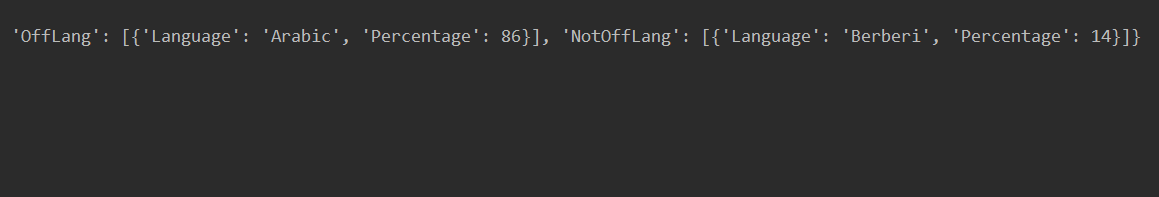
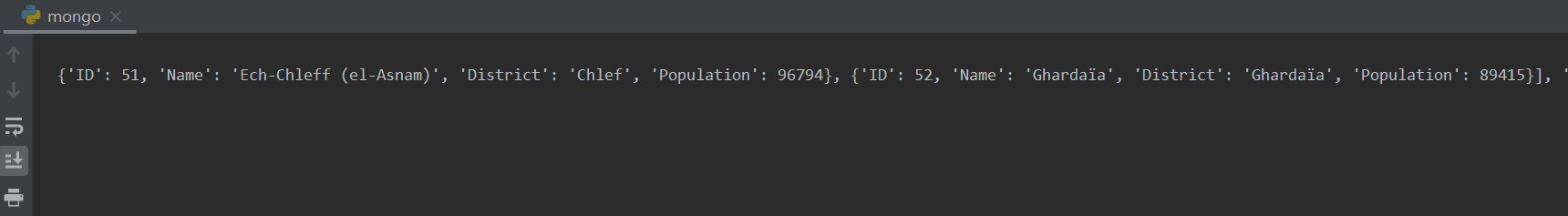
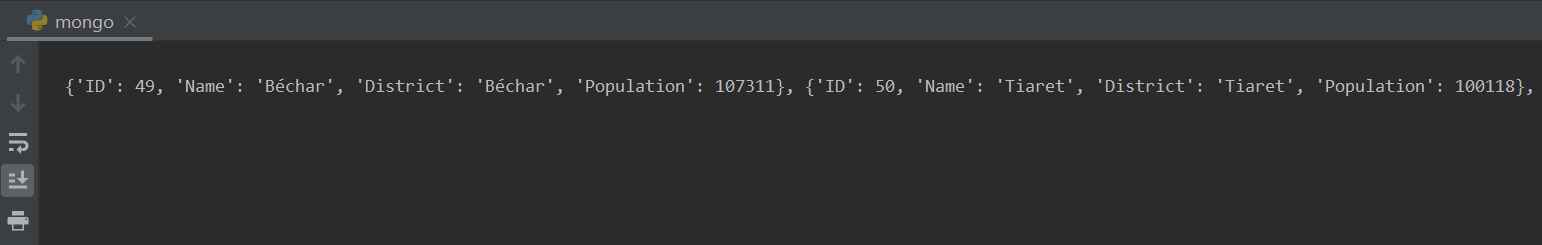
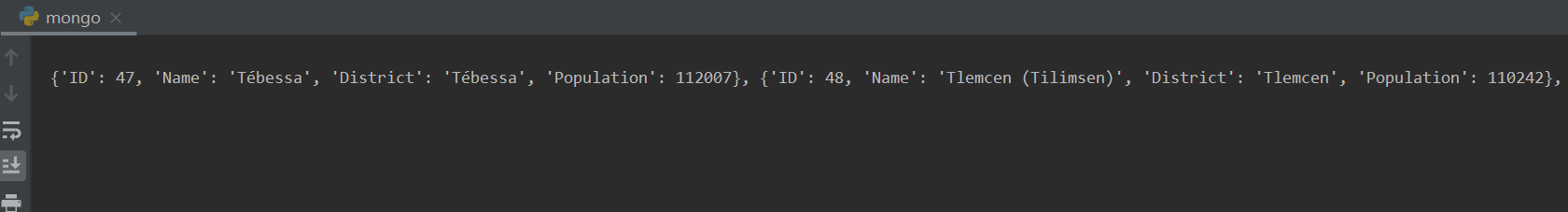
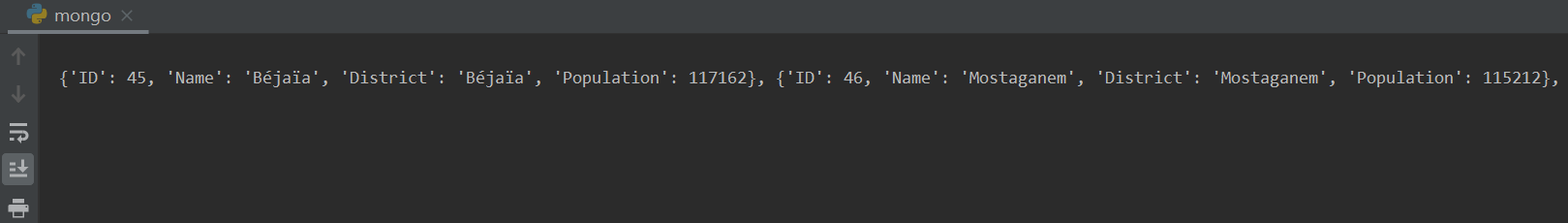
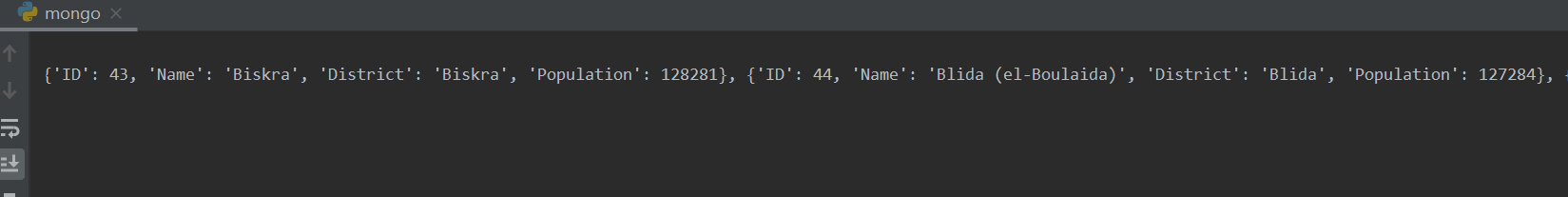
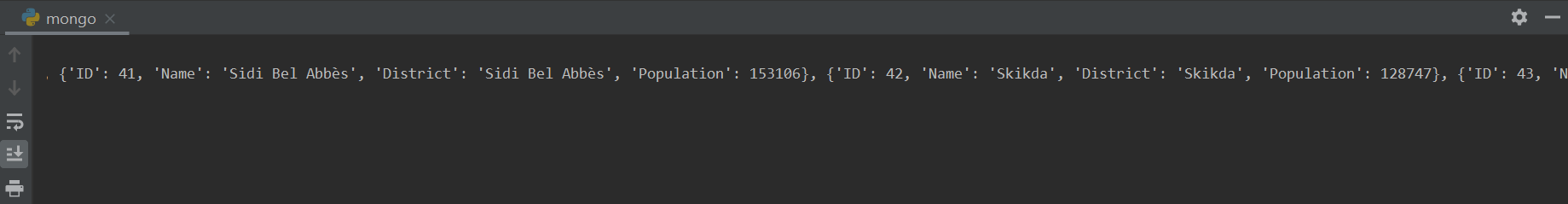
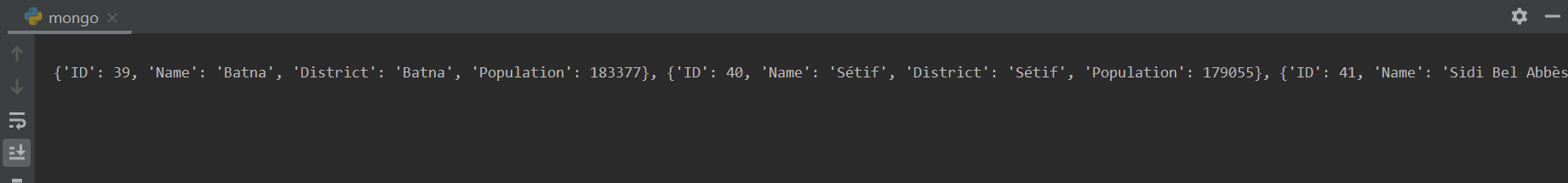
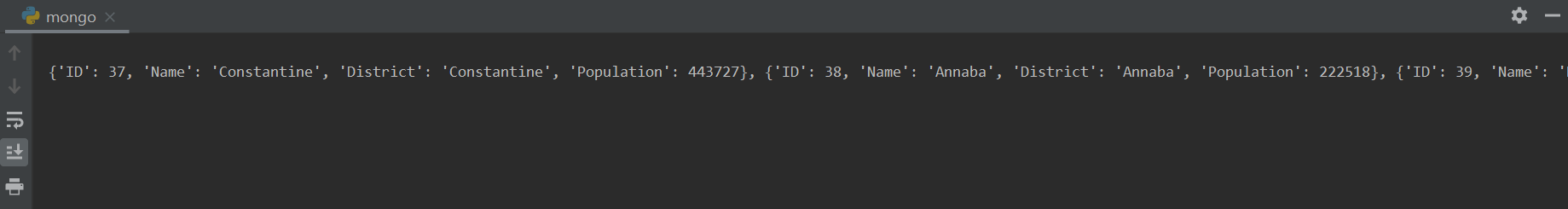
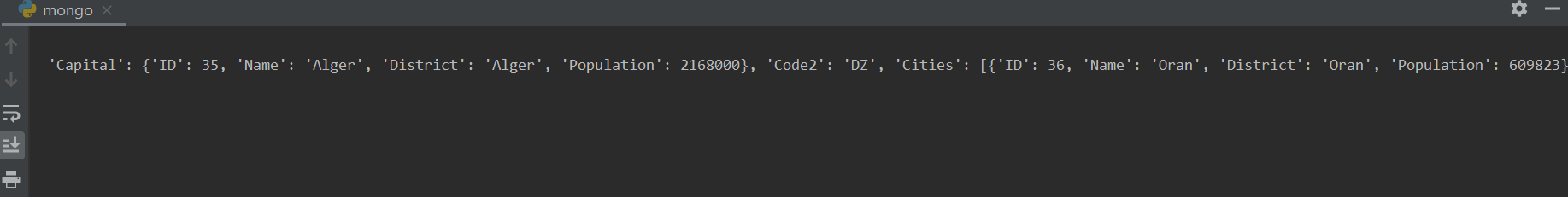
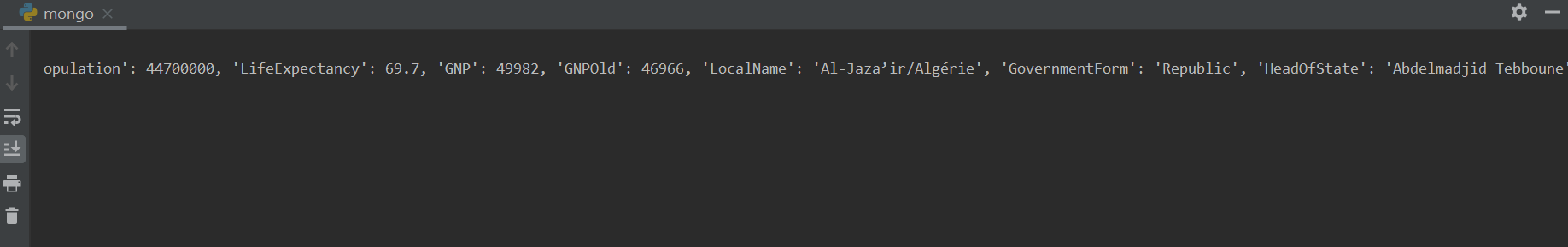
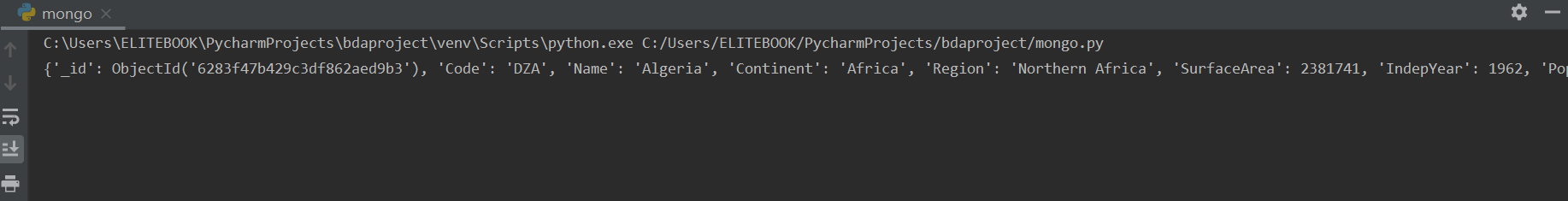
**Fonction:**

def requette3():  
 return world.find\_one({"Name":"Algeria"})

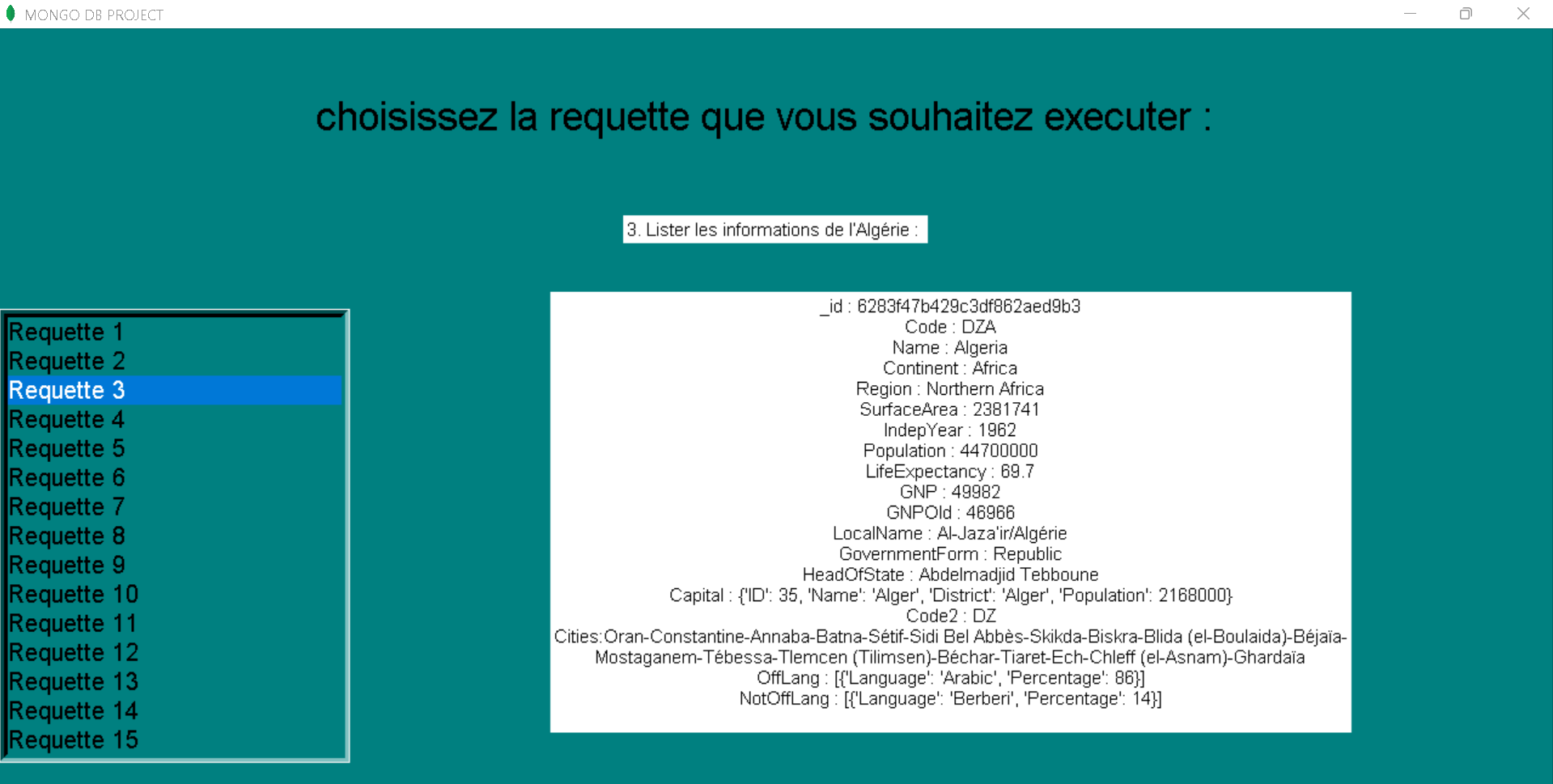
**Code:**



**Résultat sur le terminal:**



**Résultat sur l’interface :**



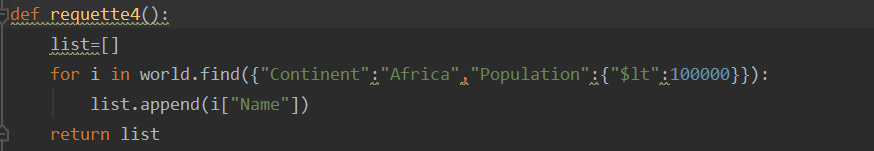
**4. Lister les pays du continent Africain, ayant une population inférieure à 100000 habitants :**

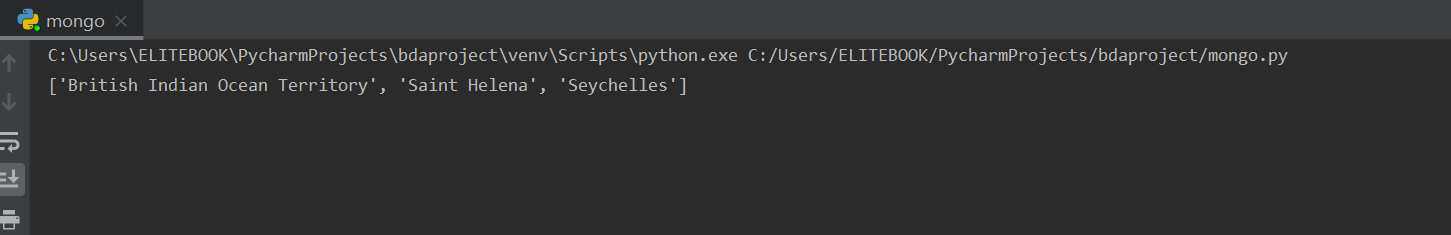
**Fonction:**

def requette4():  
 list=[]

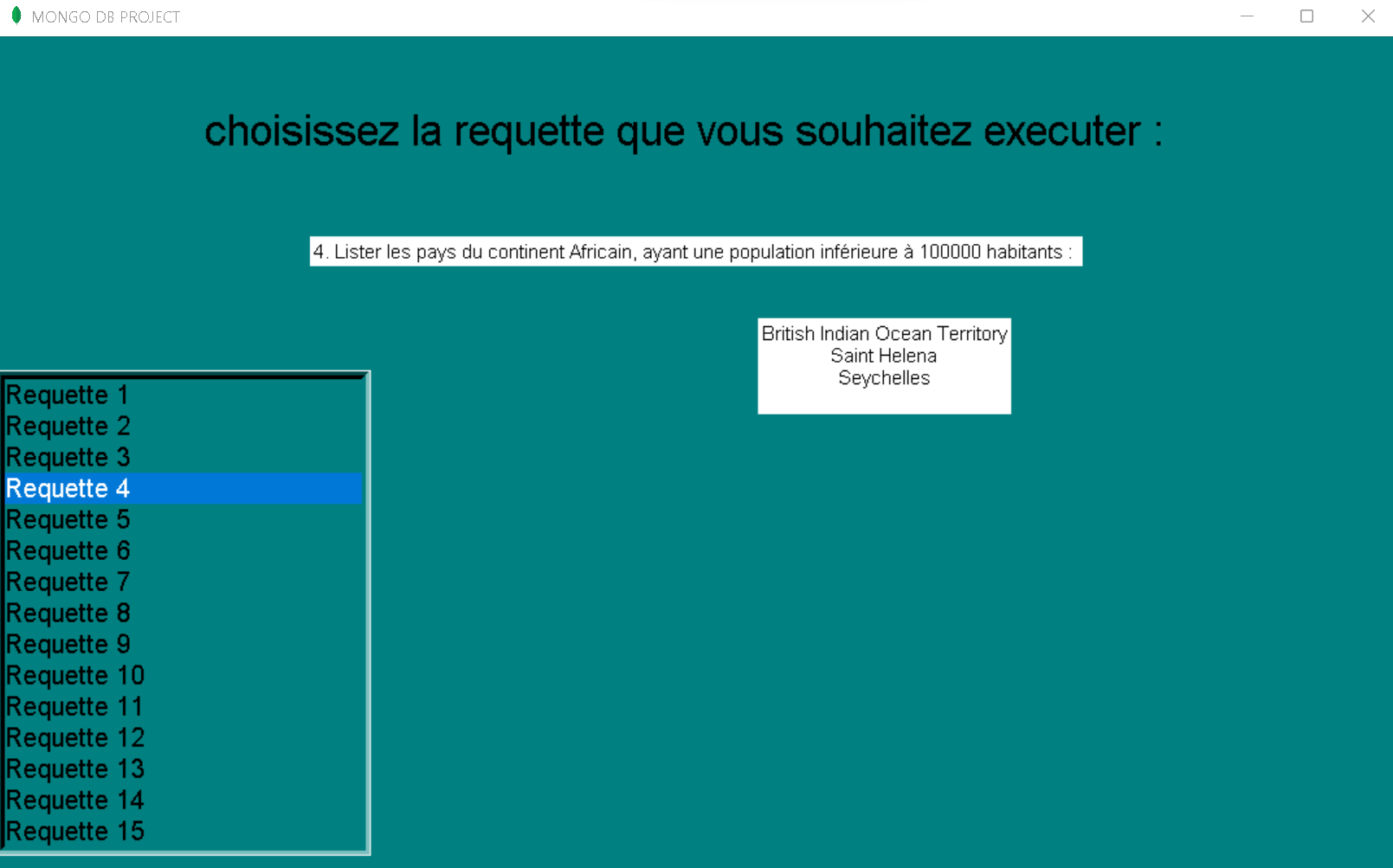
for i in world.find({"Continent":"Africa","Population": {"$lt":100000}}):  
 list.append(i["Name"])  
return list

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**

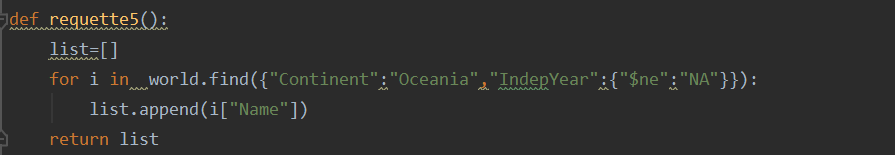


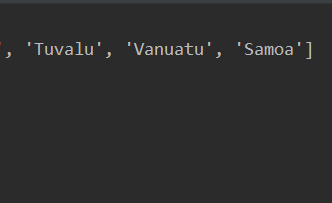
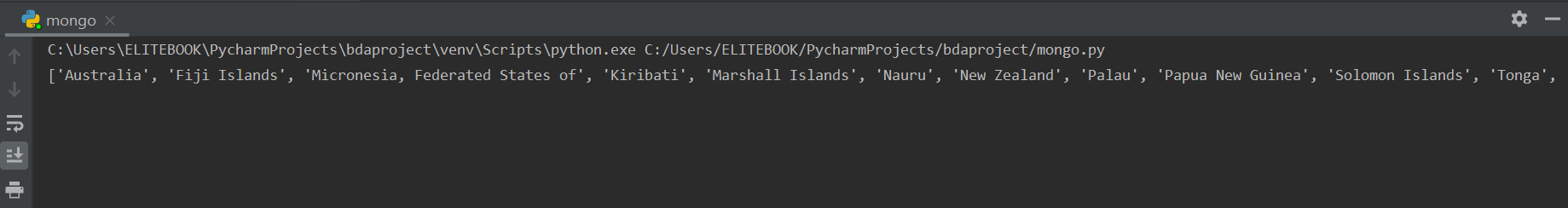
**5. Lister les pays indépendants du continent océanique :**

**Fonction:**

def requette5():  
 list=[]  
 for i in world.find({"Continent":"Oceania","IndepYear":{"$ne":"NA"}}):  
 list.append(i["Name"])  
 return list

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**

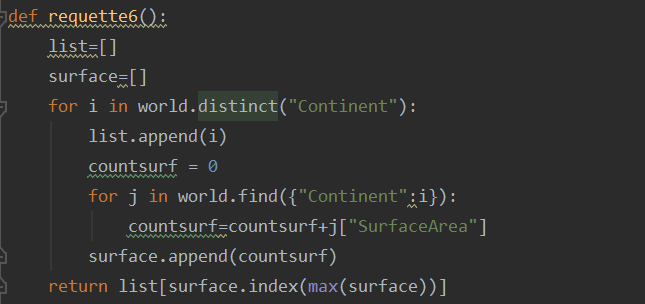


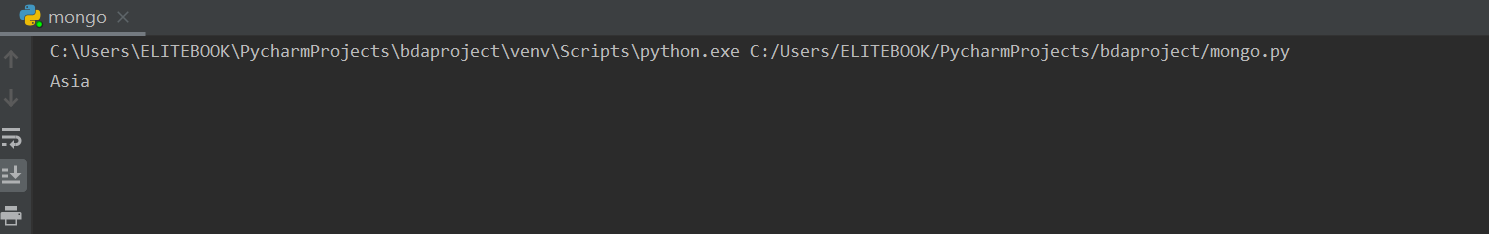
**6. plus grand continent en terme de surface :**

**Fonction:**

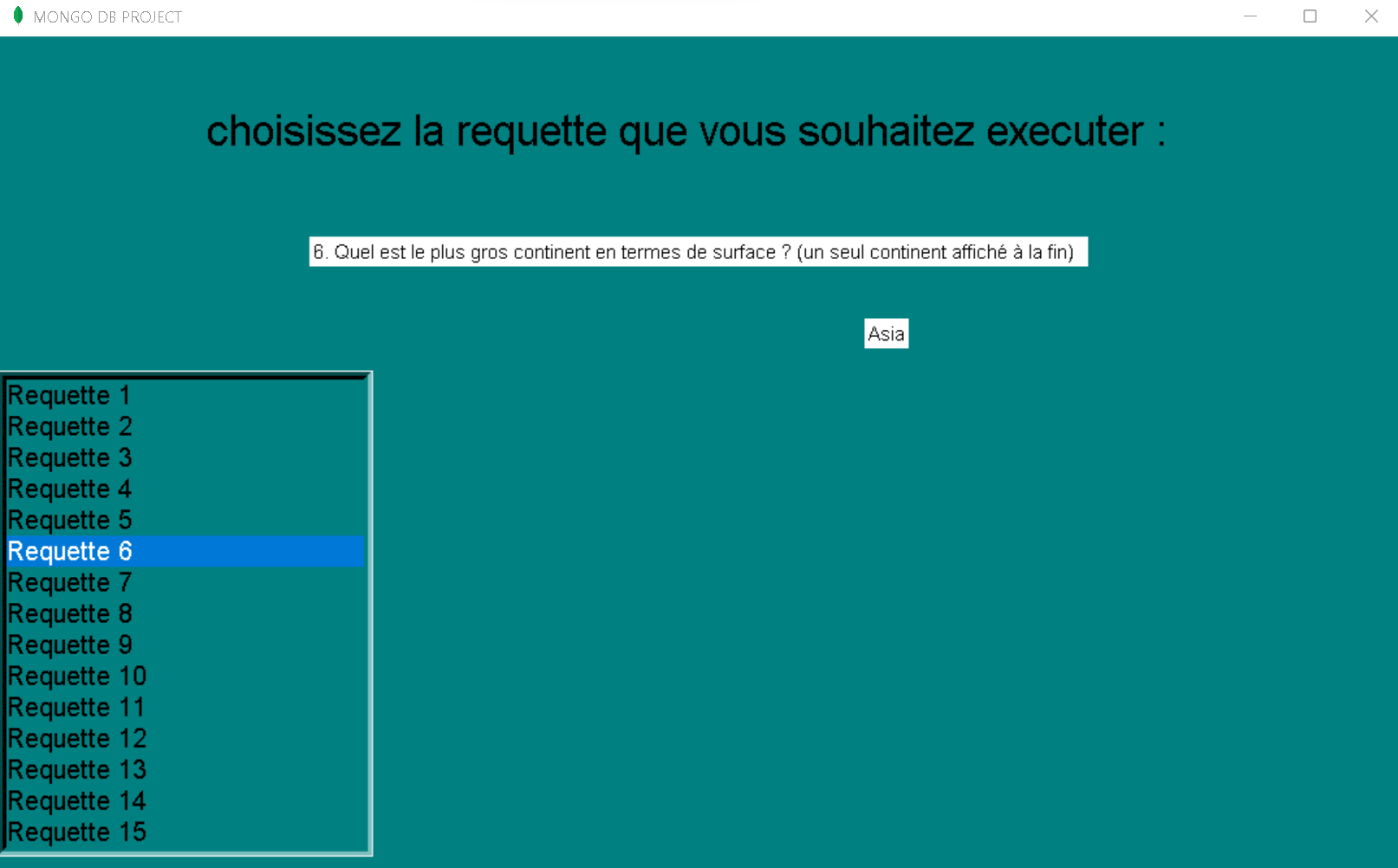
def requette6():  
 list=[]  
 surface=[]  
 for i in world.distinct("Continent"):  
 list.append(i)  
 countsurf = 0  
 for j in world.find({"Continent":i}):  
 countsurf=countsurf+j["SurfaceArea"]  
 surface.append(countsurf)  
 return list[surface.index(max(surface))]

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**

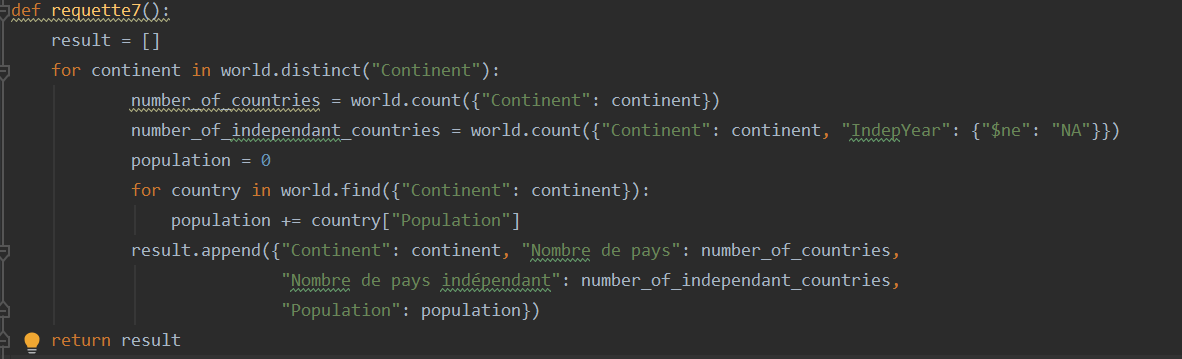


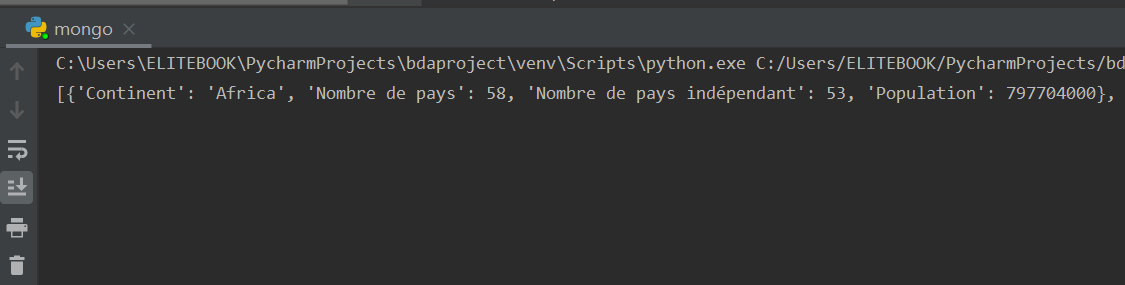
**7. Donner par continent le nombre de pays , nombre de pays indépendants et population :**

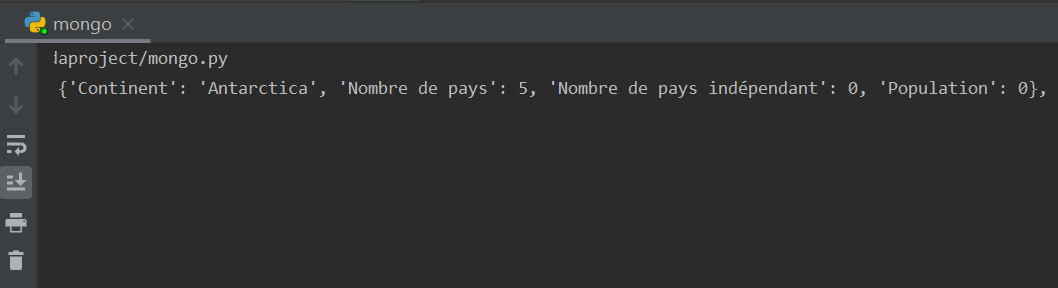
**Fonction:**

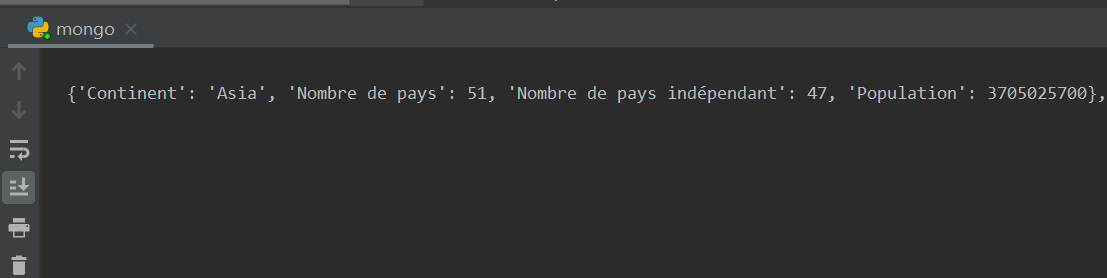
def requette7():  
 result = []  
 for continent in world.distinct("Continent"):  
 number\_of\_countries = world.count({"Continent": continent})  
 number\_of\_independant\_countries = world.count({"Continent": continent, "IndepYear": {"$ne": "NA"}})  
 population = 0  
 for country in world.find({"Continent": continent}):  
 population += country["Population"]  
 result.append({"Continent": continent, "Nombre de pays": number\_of\_countries,"Nombre de pays indépendant": number\_of\_independant\_countries,"Population": population})  
 return result

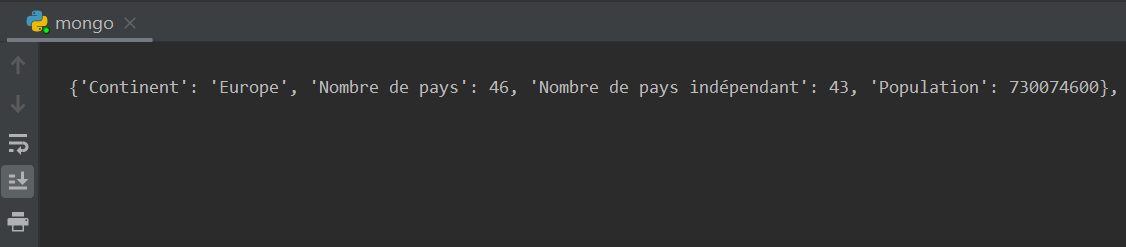
**Code:**

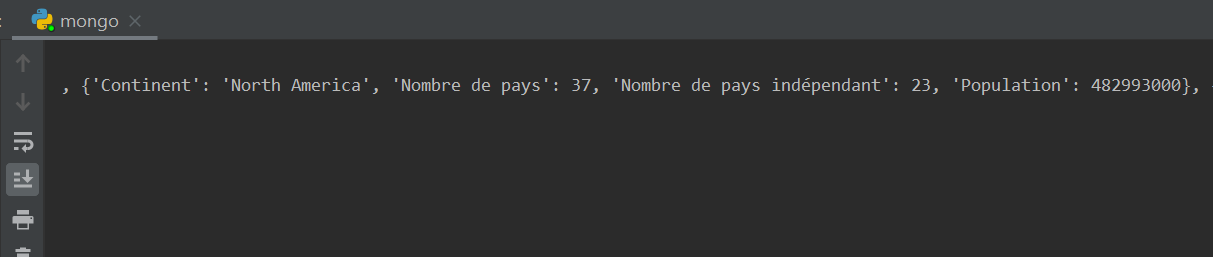


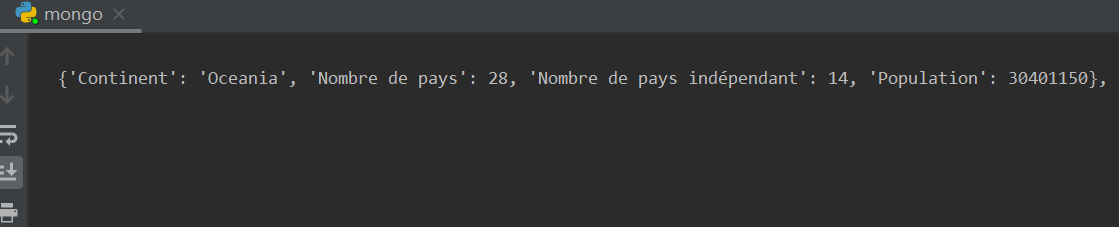
**Résultat sur le terminal:**

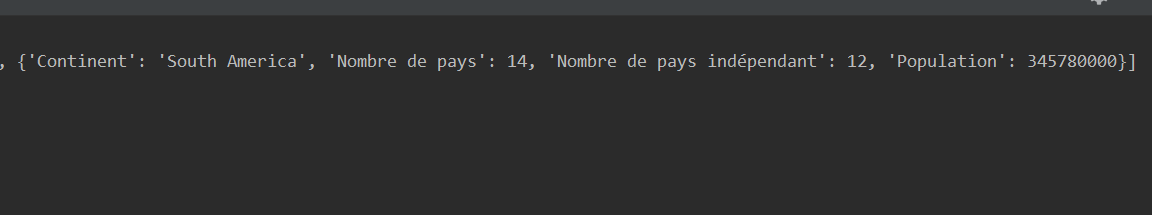




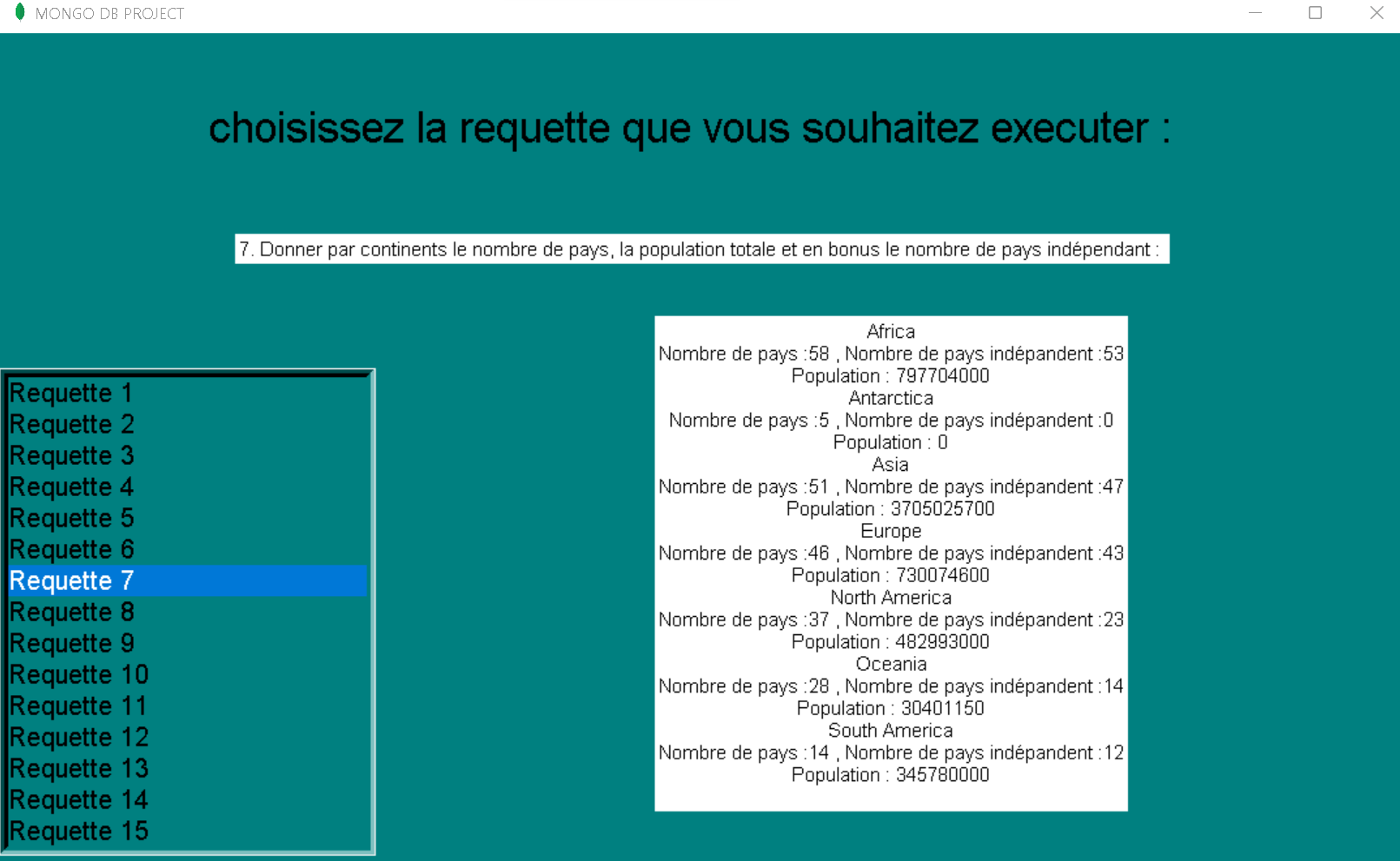








**Résultat sur l’interface :**

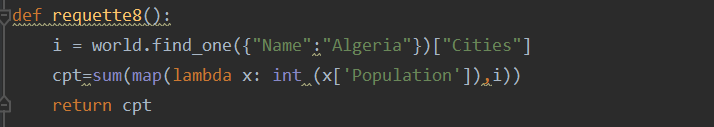


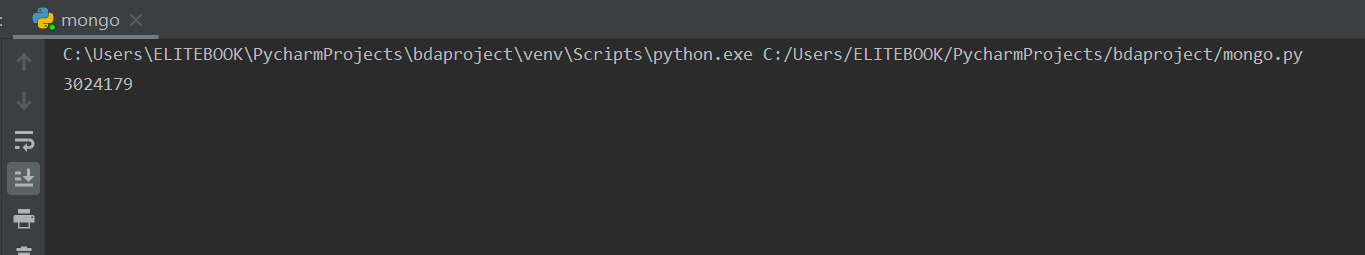
**8. Donner la population totale des villes d’Algérie :**

**Fonction:**

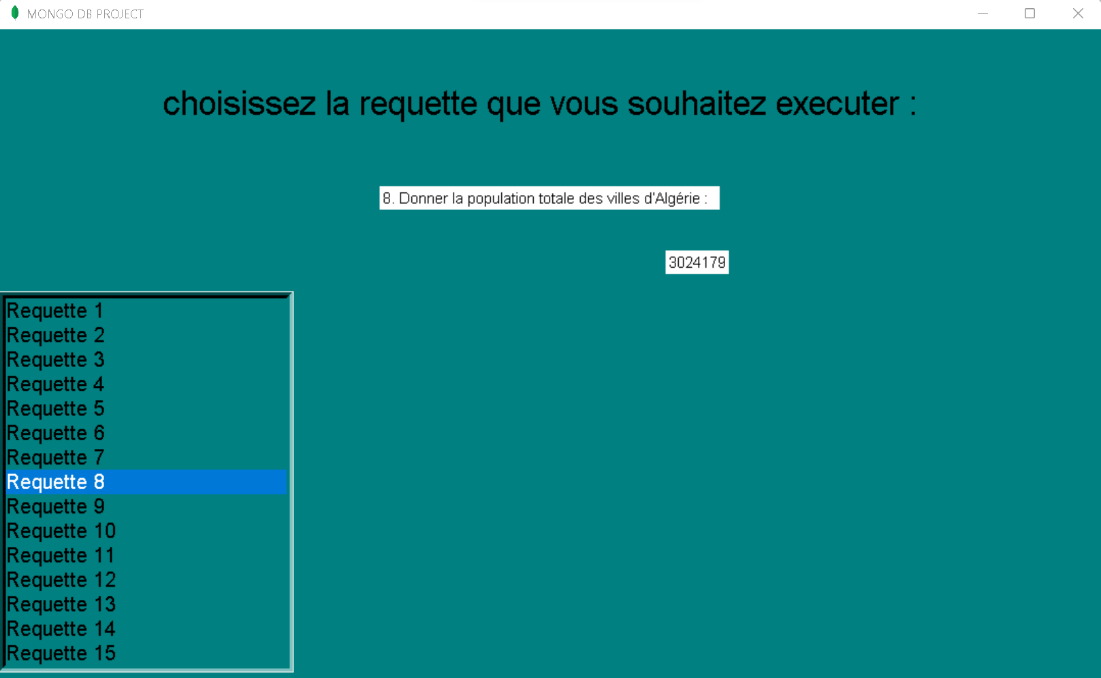
def requette8():  
 i = world.find\_one({"Name":"Algeria"})["Cities"]  
 cpt=sum(map(lambda x: int (x['Population']),i))  
 return cpt

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**

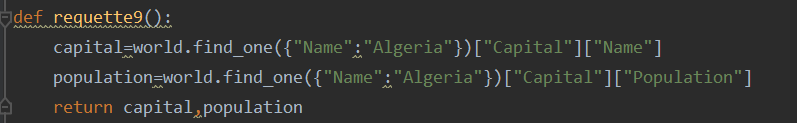


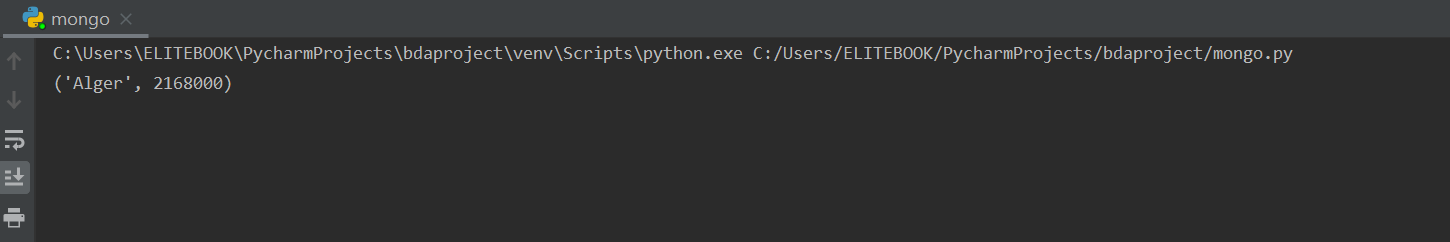
**9. Donner la capitale d’Algérie et sa population :**

**Fonction:**

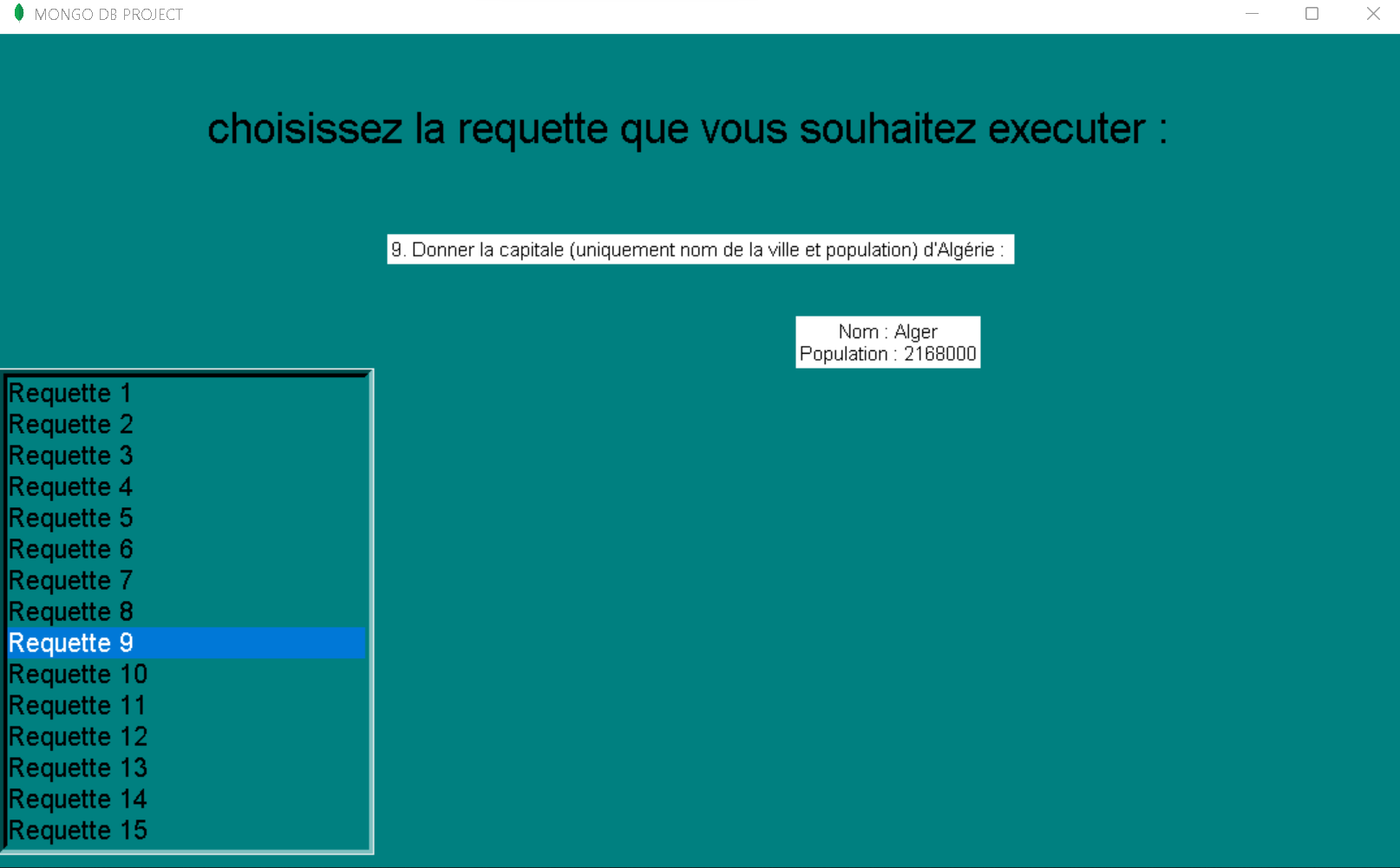
def requette9():  
 capital=world.find\_one({"Name":"Algeria"})["Capital"]["Name"]  
 population=world.find\_one({"Name":"Algeria"})["Capital"]["Population"]  
 return capital,population

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**

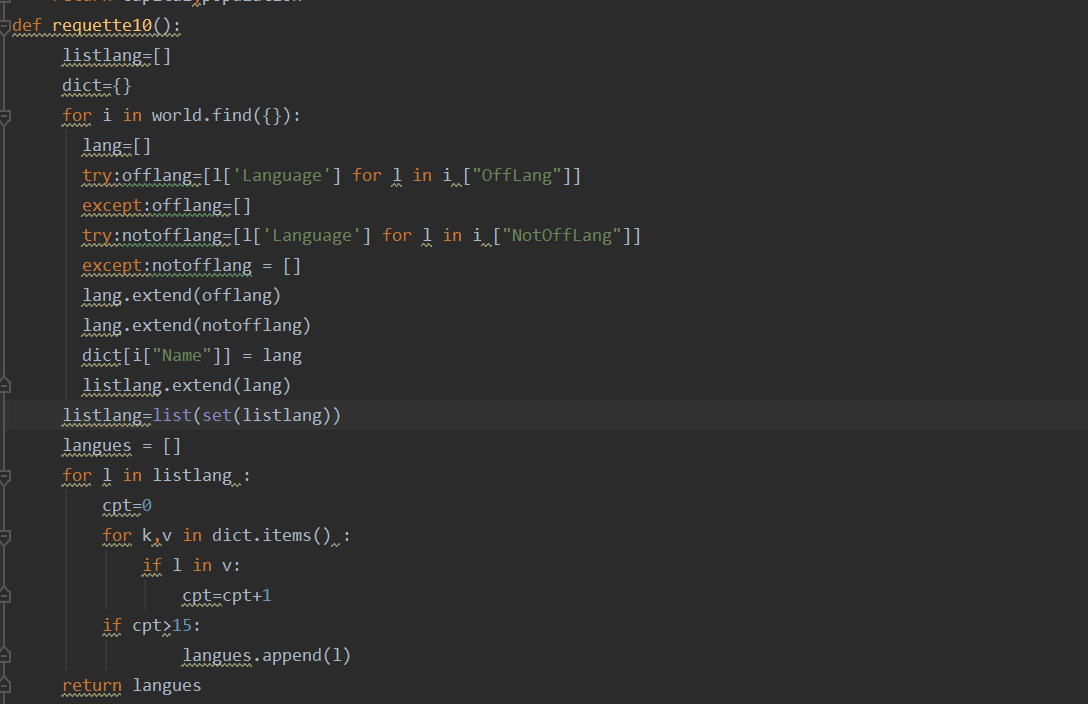


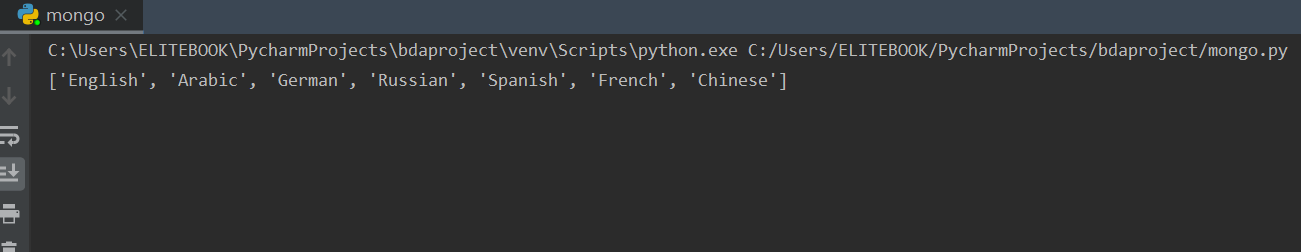
**10. Les langues parlées dans plus de 15 pays :**

**Fonction:**

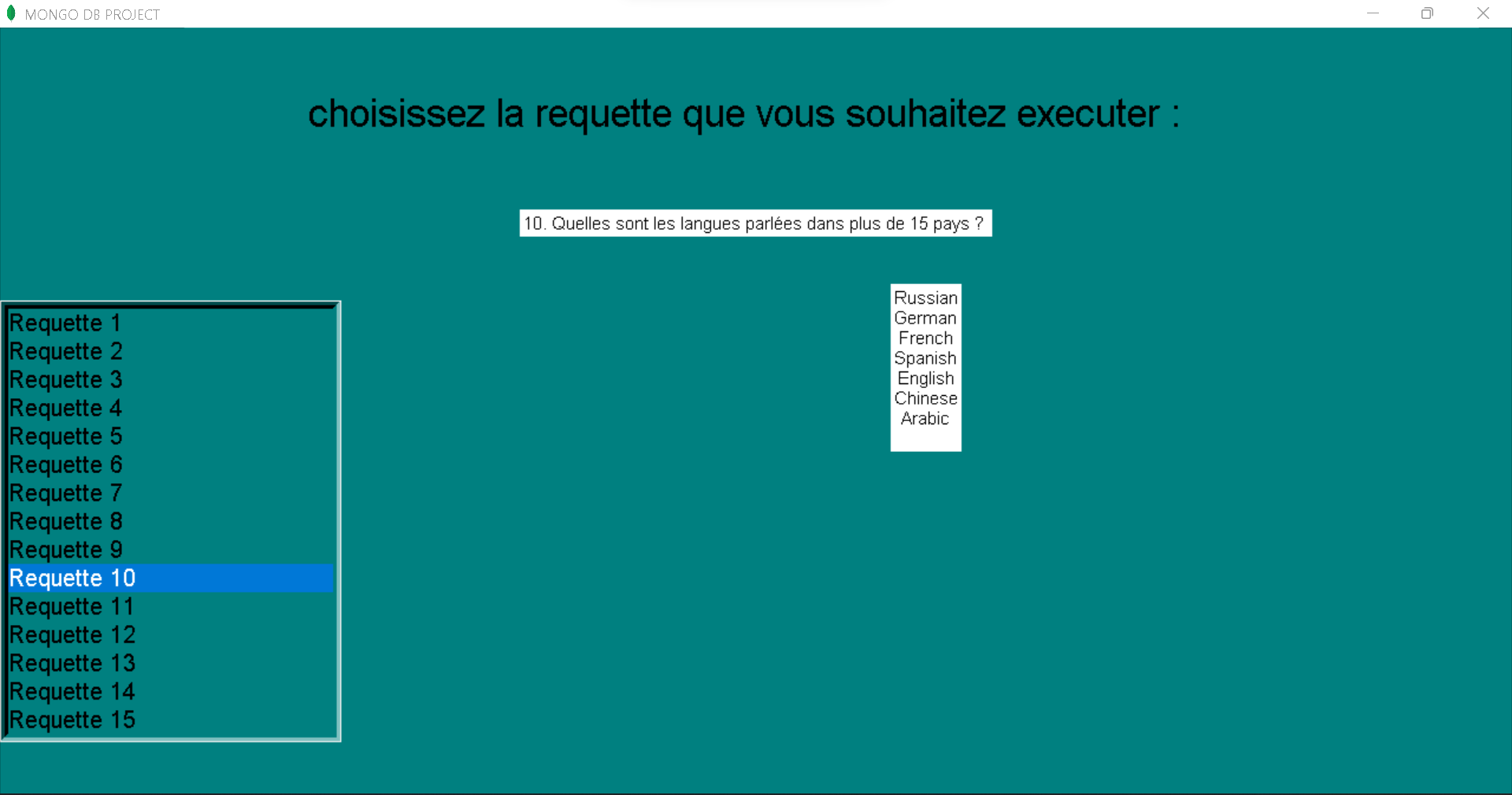
def requette10():  
 listlang=[]  
 dict={}  
 for i in world.find({}):  
 lang=[]  
 try:  
 offlang=[l['Language'] for l in i ["OffLang"]]  
 except:  
 offlang=[]  
 try:  
 notofflang=[l['Language'] for l in i ["NotOffLang"]]  
 except:  
 notofflang = []  
 lang.extend(offlang)  
 lang.extend(notofflang)  
 dict[i["Name"]] = lang  
 listlang.extend(lang)  
 listlang=list(set(listlang))  
 langues=[]  
 for l in listlang :  
 cpt=0  
 for k,v in dict.items() :  
 if l in v:  
 cpt=cpt+1  
 if cpt>15:  
 langues.append(l)  
 return langues

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**



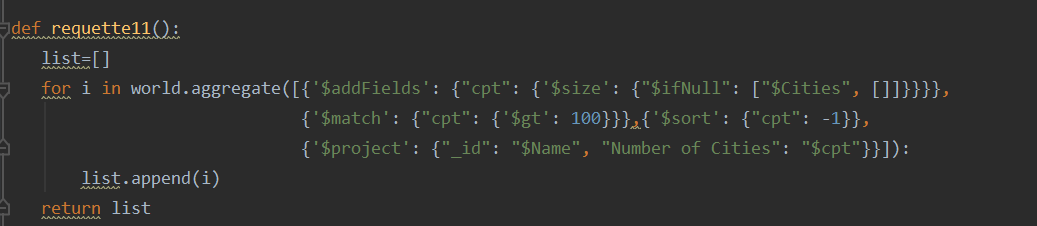
pour voir aussi la methode d’aggregation on a basculer de la methode algorithmique à la methode aggregate pour les prochaines requettes .

**11. Calculer pour chaque pays le nombre de villes (pour les pays ayant au moins 100 villes ) en les trient par ordre décroissant du nombre des villes :**

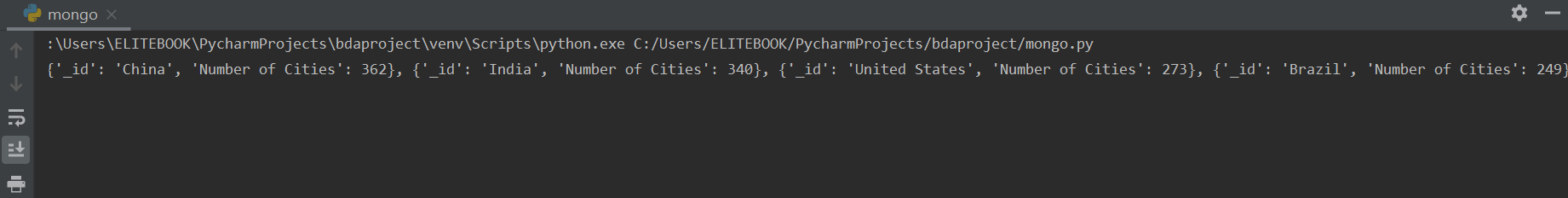
**Fonction:**

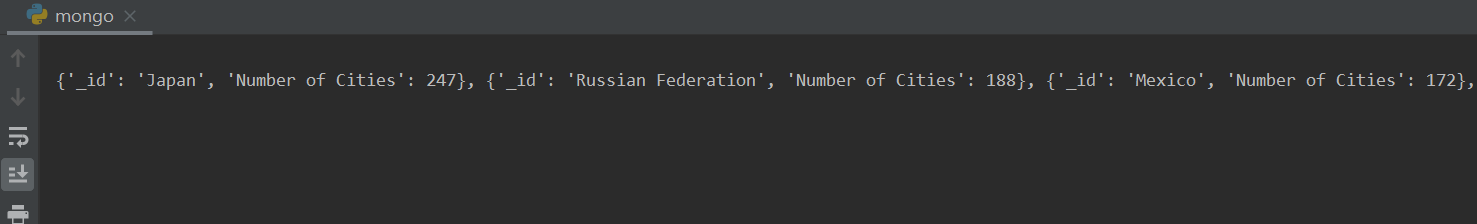
def requette11():  
 list=[]  
 for i in world.aggregate([{'$addFields': {"cpt": {'$size': {"$ifNull": ["$Cities", []]}}}},  
 {'$match': {"cpt": {'$gt': 100}}},{'$sort': {"cpt": -1}},  
 {'$project': {"\_id": "$Name", "Number of Cities": "$cpt"}}]):  
 list.append(i)  
 return list

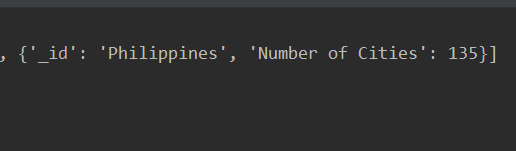
**Code:**



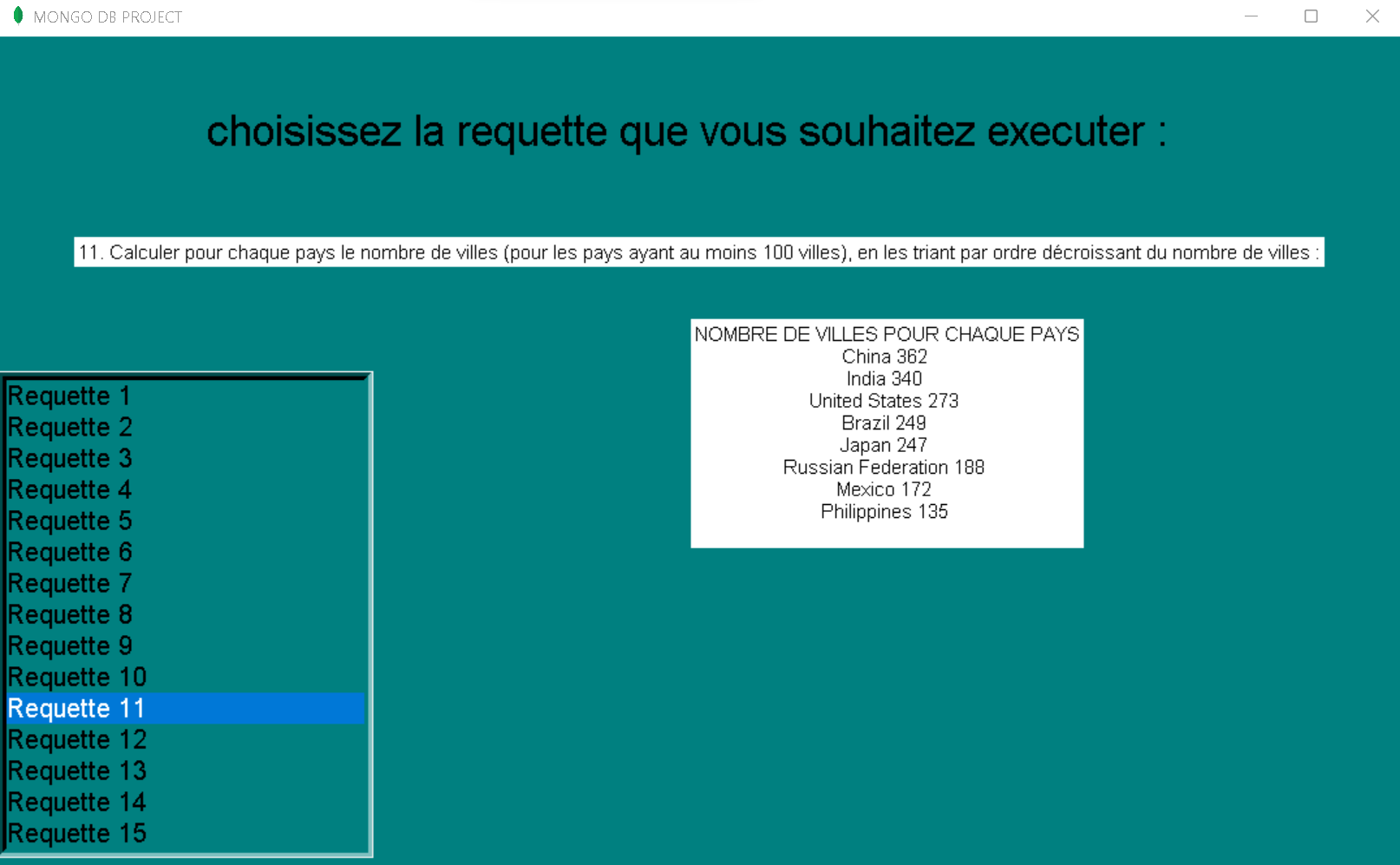
**Résultat sur le terminal:**







**Résultat sur l’interface :**



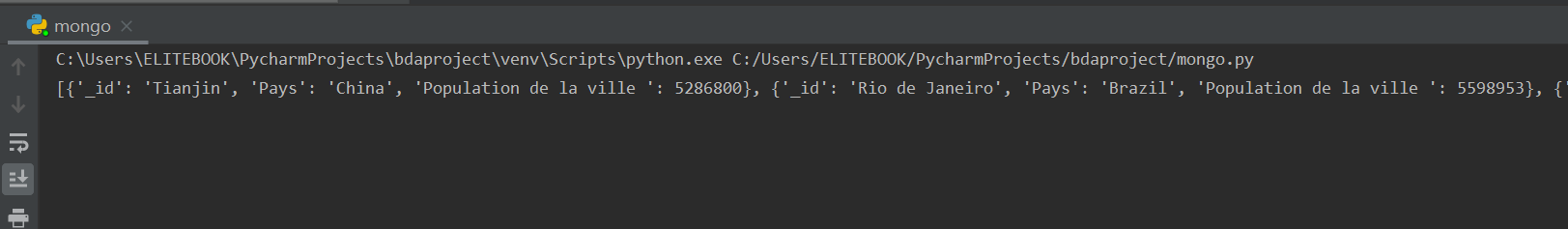
**12.Lister les 10 villes les plus habitées ainsi que leur pays dans l’ordre décroissant de la population :**

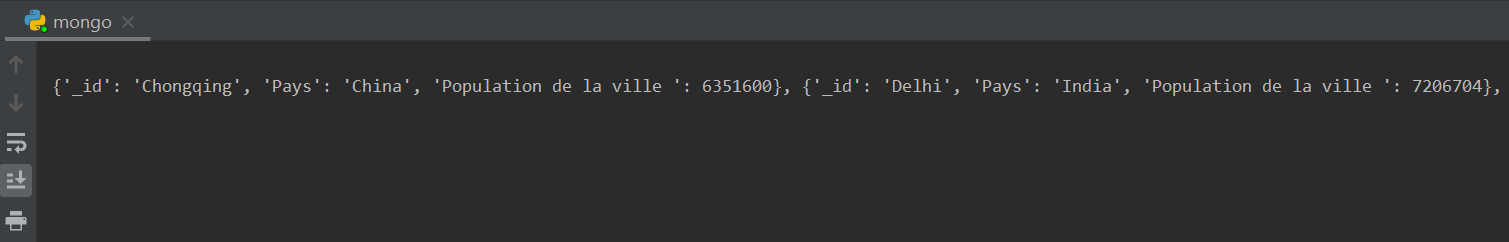
def requette12():  
 list=[]  
 for i in world.aggregate([{'$unwind': "$Cities"}, {'$sort': {"Cities.Population": -1}}, {'$limit': 10},  
 {'$sort': {"Cities.Population": 1}},  
 {'$project': {"\_id": "$Cities.Name", "Pays": "$Name", "Population de la ville ": "$Cities.Population"}}]):  
 list.append(i)  
 return list

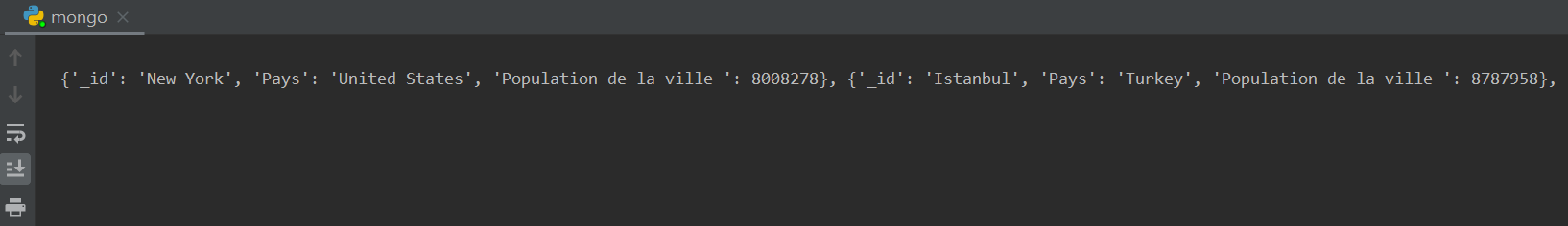
**Code:**

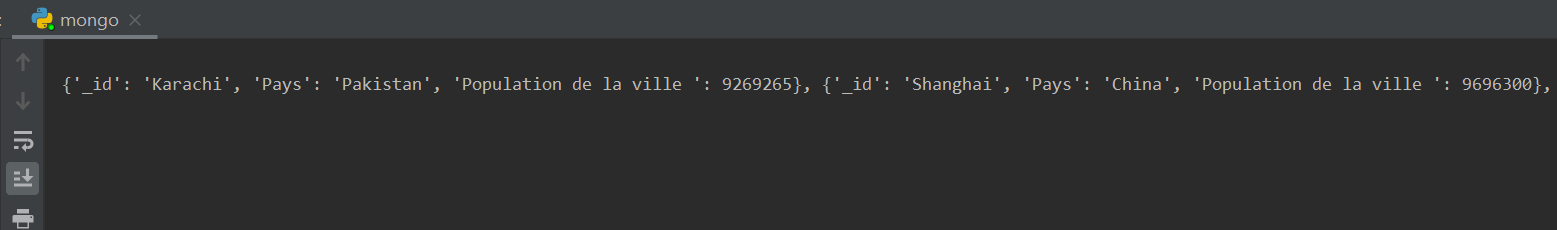


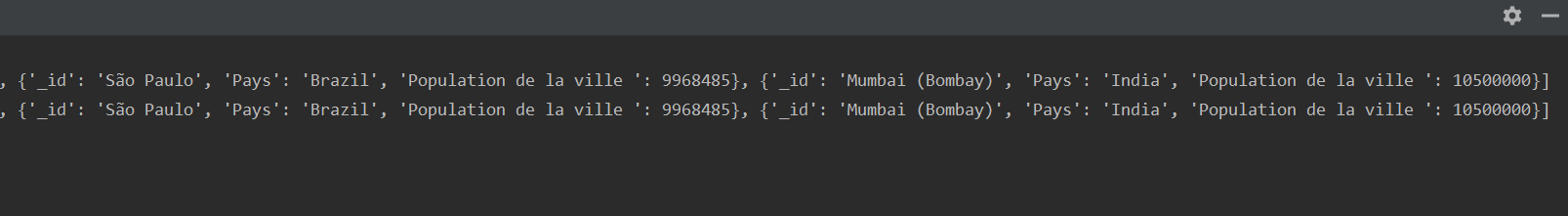
**Résultat sur le terminal:**



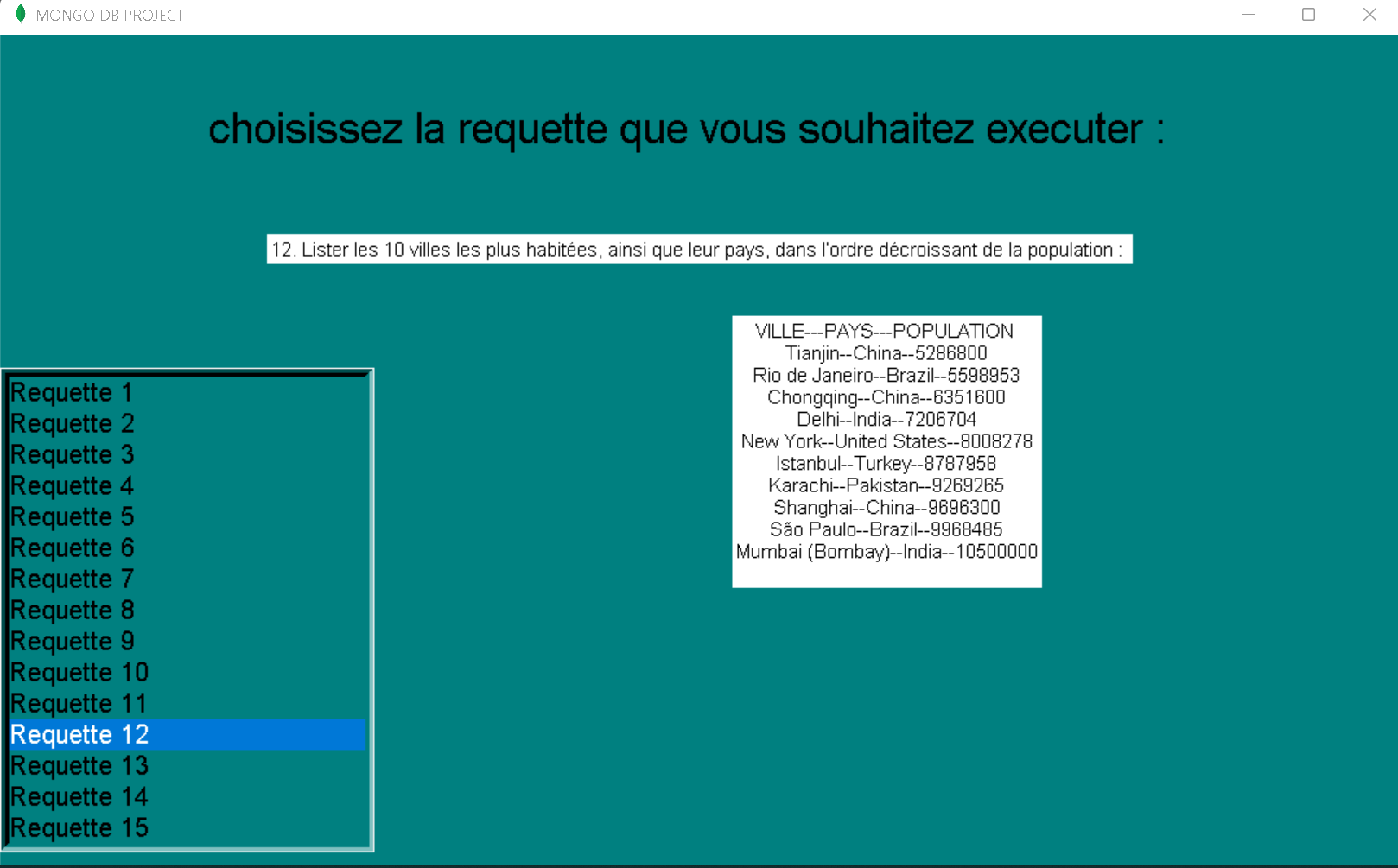








**Résultat sur l’interface :**

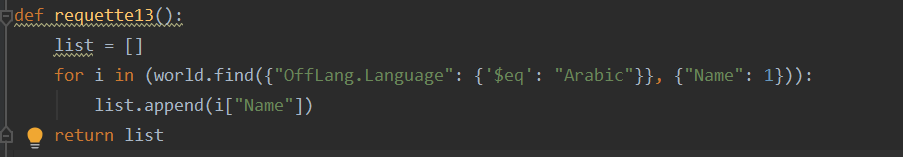


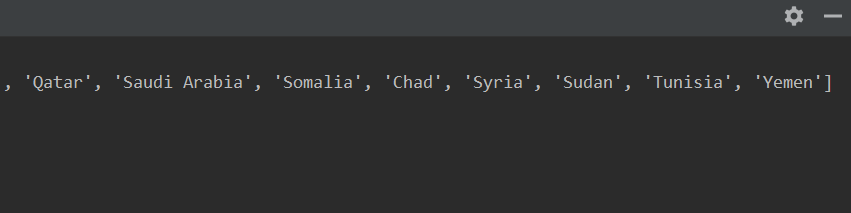
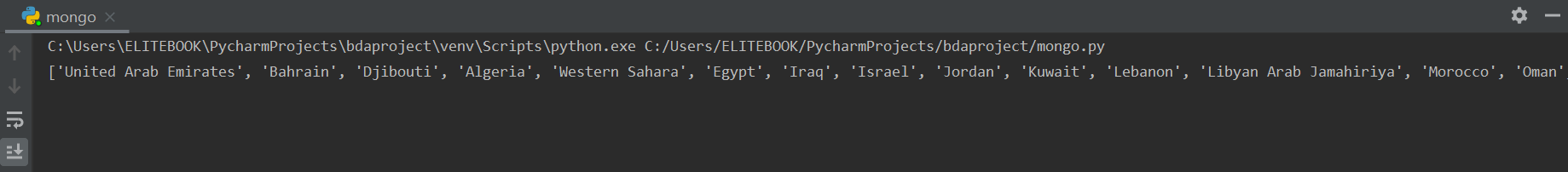
**13. Lister les pays pour lequels l’Arabe est la langue officielle :**

**Fonction:**

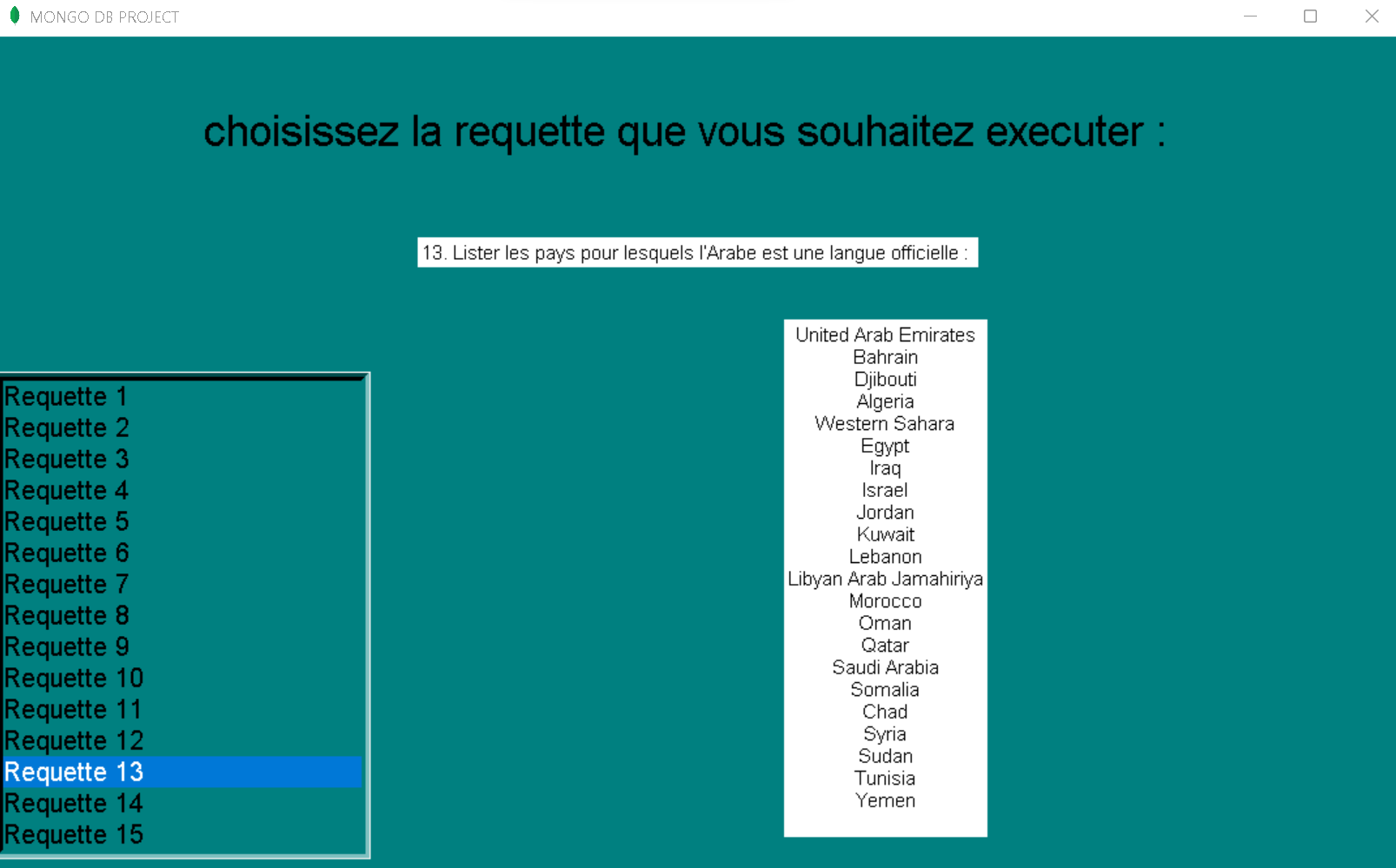
def requette13():  
 list = []  
 for i in (world.find({"OffLang.Language": {'$eq': "Arabic"}}, {"Name": 1})):  
 list.append(i["Name"])  
 return list

**Code:**



**Résultat sur le terminal:**

**Résultat sur l’interface :**

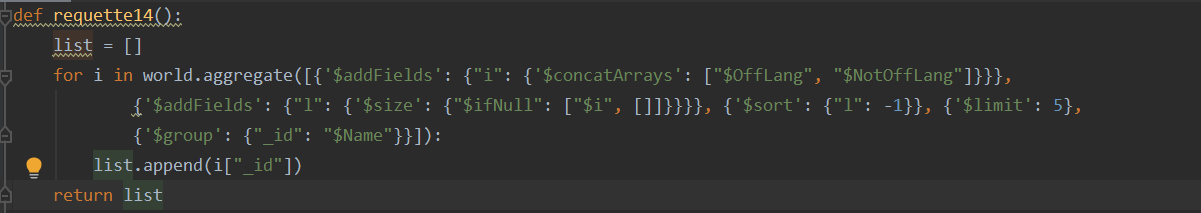


**14.Lister les 5 pays avec le plus de langues parlées :**

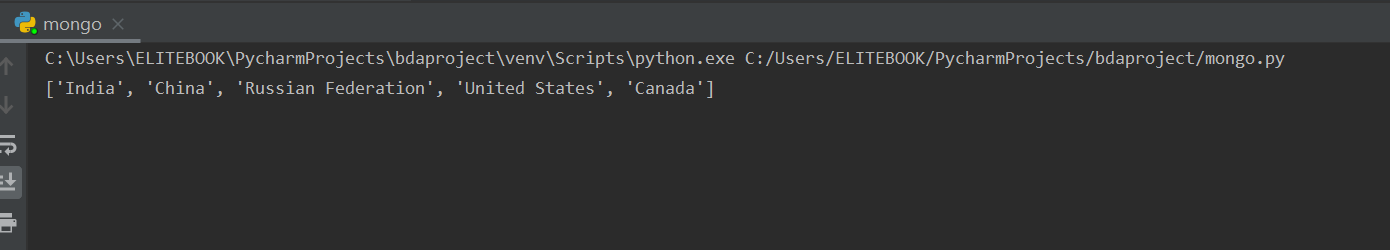
**Fonction :**

def requette14():  
 list = []  
 for i in world.aggregate([{'$addFields': {"i": {'$concatArrays': ["$OffLang", "$NotOffLang"]}}},  
 {'$addFields': {"l": {'$size': {"$ifNull": ["$i", []]}}}}, {'$sort': {"l": -1}}, {'$limit': 5},  
 {'$group': {"\_id": "$Name"}}]):  
 list.append(i["\_id"])  
 return list

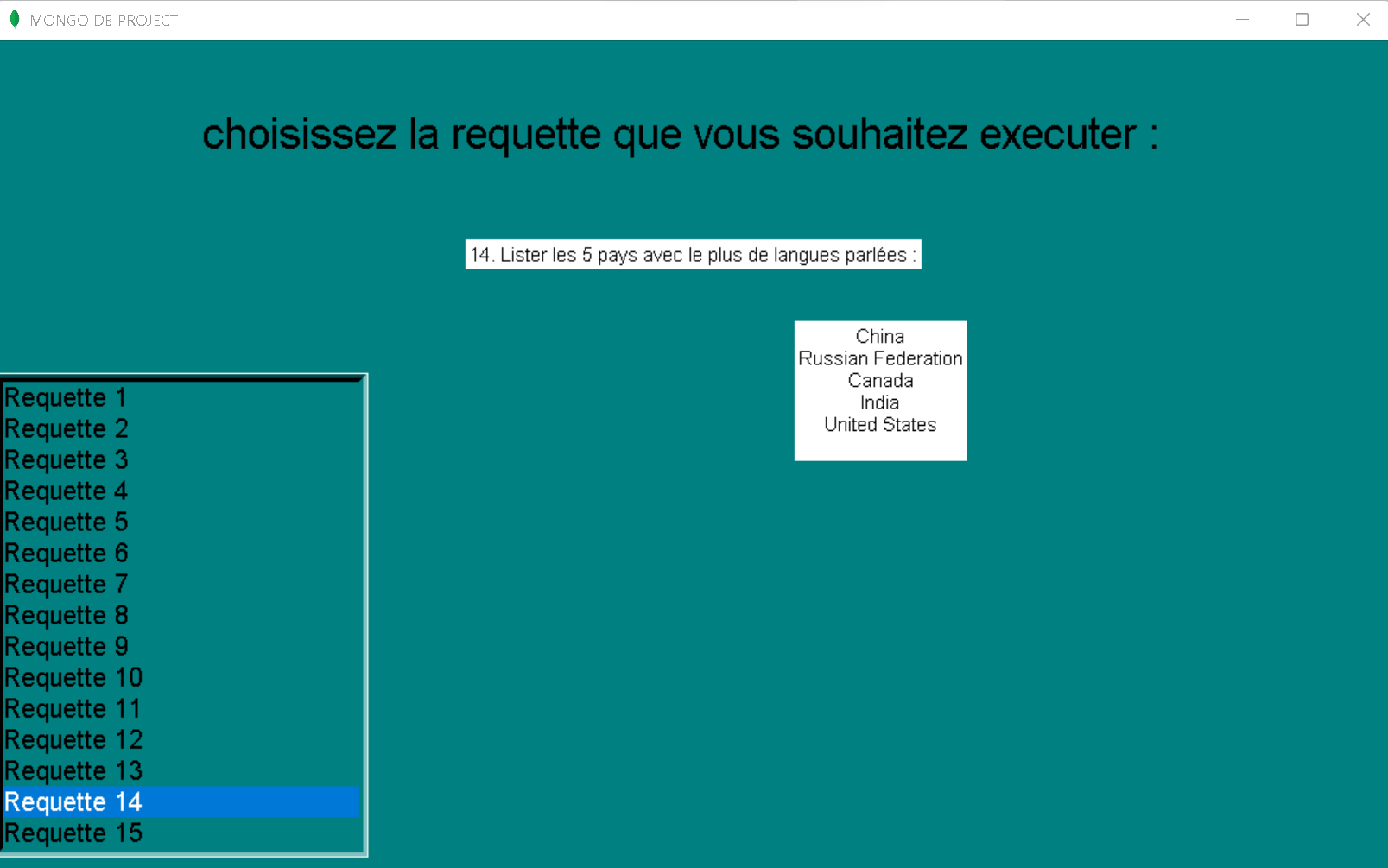
**Code:**



**Résultat sur le terminal:**



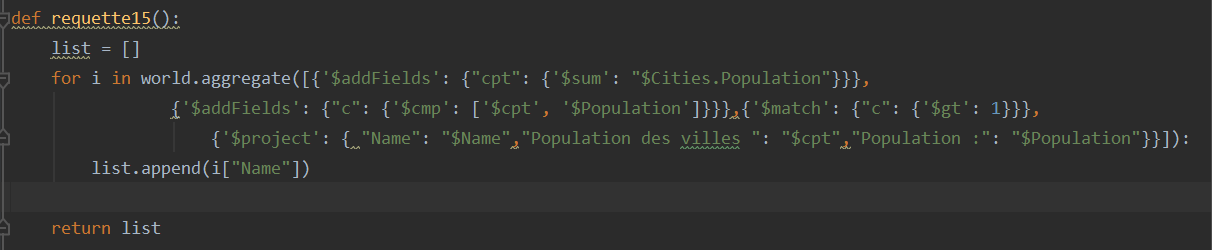
**Résultat sur l’interface :**



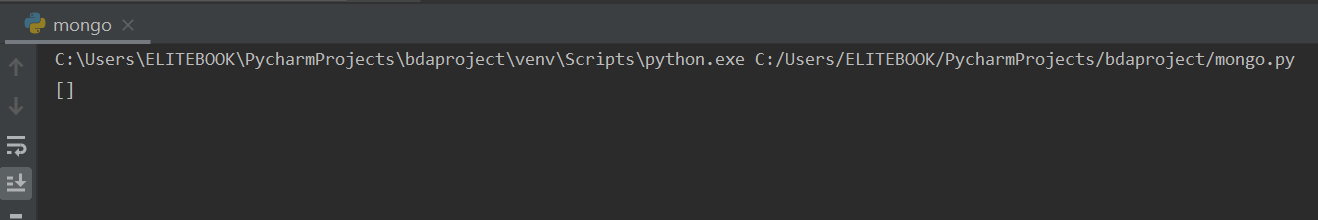
**15.Lister les pays pour lesquels la somme des populations des villes est supérieure à la population du pays :**

def requette15():  
 list = []  
 for i in world.aggregate([{'$addFields': {"cpt": {'$sum': "$Cities.Population"}}},  
 {'$addFields': {"c": {'$cmp': ['$cpt', '$Population']}}},{'$match': {"c": {'$gt': 1}}},  
 {'$project': { "Name": "$Name","Population des villes ": "$cpt","Population :": "$Population"}}]):  
 list.append(i["Name"])  
  
 return list

**Code:**



**Résultat sur le terminal:**



**Résultat sur l’interface :**

