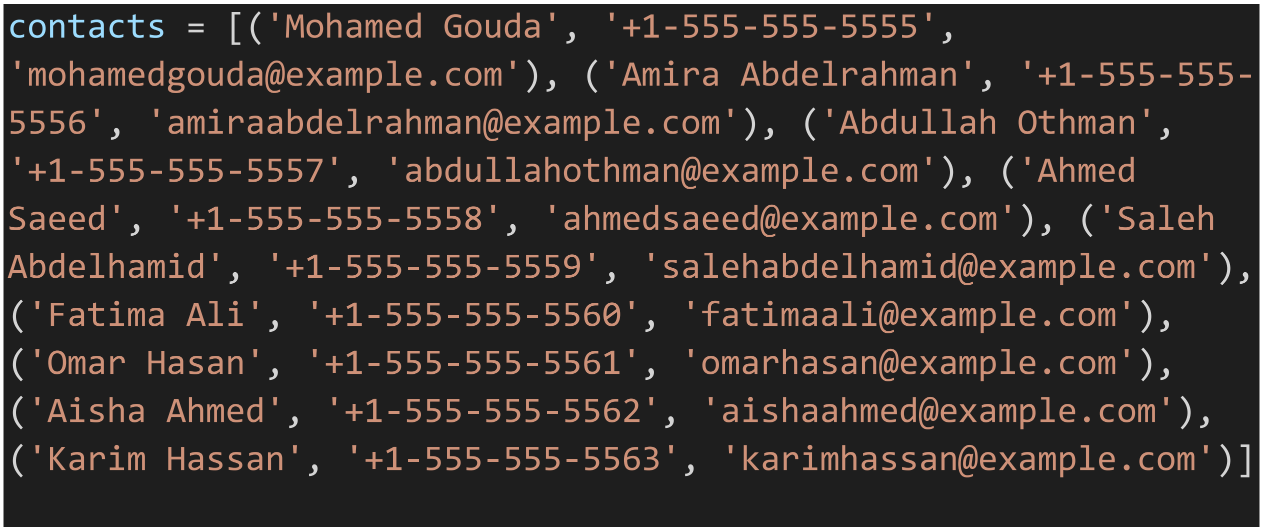
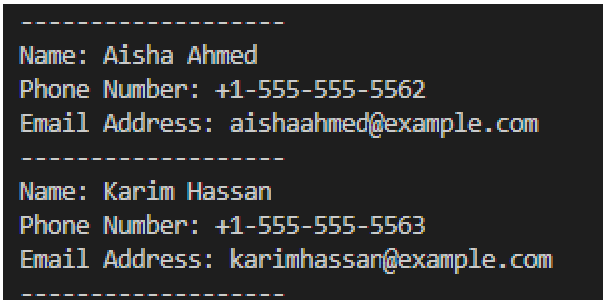
1.قم بعمل برنامج يقوم بعرض معلومات جهات الاتصال التالية بالشكل التالي

Make a program that print contacts info in the following form.



```
contacts = [('Mohamed Gouda', '+1-555-555-5555',
'mohamedgouda@example.com'), ('Amira Abdelrahman', '+1-555-555-
5556', 'amiraabdelrahman@example.com'), ('Abdullah Othman',
'+1-555-555-5557', 'abdullahothman@example.com'), ('Ahmed
Saeed', '+1-555-555-5558', 'ahmedsaeed@example.com'), ('Saleh
Abdelhamid', '+1-555-555-5559', 'salehabdelhamid@example.com'),
('Fatima Ali', '+1-555-555-5560', 'fatimaali@example.com'),
('Omar Hasan', '+1-555-555-5561', 'omarhasan@example.com'),
('Aisha Ahmed', '+1-555-555-5562', 'aishaahmed@example.com'),
('Karim Hassan', '+1-555-555-5563', 'karimhassan@example.com')]
```



```
--------------------
Name: Aisha Ahmed
Phone Number: +1-555-555-5562
Email Address: aishaahmed@example.com
--------------------
Name: Karim Hassan
Phone Number: +1-555-555-5563
Email Address: karimhassan@example.com
--------------------
```

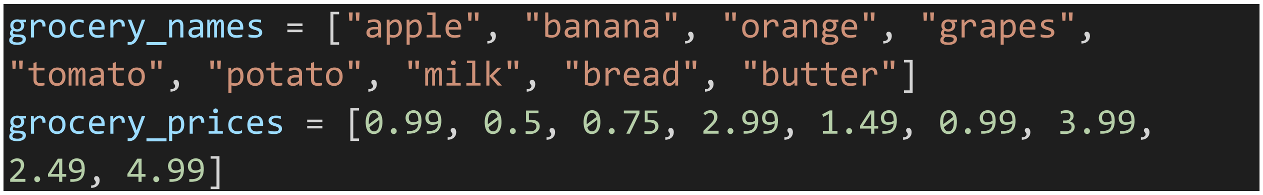
إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# 1. Program to store and display contact information using tuples

# A list of tuples to store the contact information

# Display the contact information

# Loop through the list of tuples and display the contact information
```

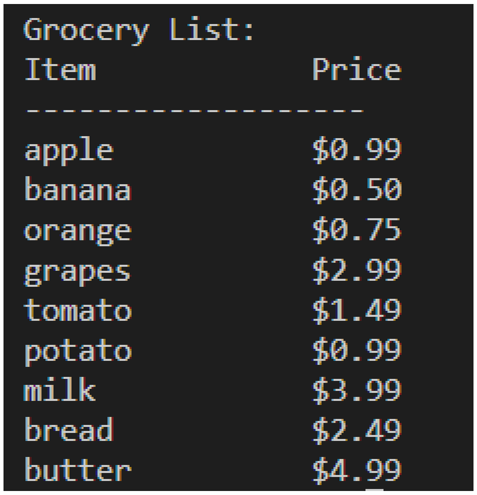
```python
# Display the contact information
print("Contact Information")
print("--------------------")

# Loop through the list of tuples and display the contact information
for contact in contacts:
    print(f"Name: {contact[0]}")
    print(f"Phone Number: {contact[1]}")
    print(f"Email Address: {contact[2]}")
    print("--------------------")
```

2. قم بعمل برنامج يقوم بطباعة القوائم التالية بالشكل التالي

Make a program that produce the following output.



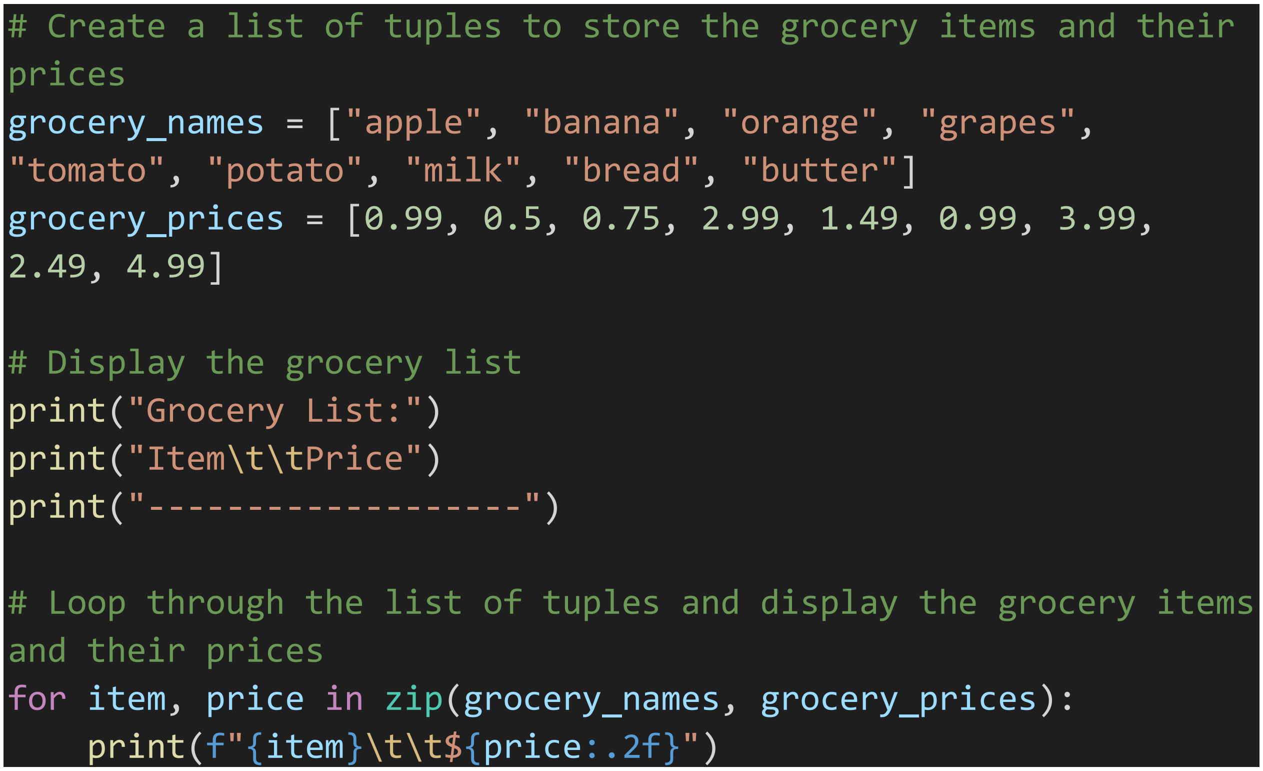
```
grocery_names = ["apple", "banana", "orange", "grapes",
"tomato", "potato", "milk", "bread", "butter"]
grocery_prices = [0.99, 0.5, 0.75, 2.99, 1.49, 0.99, 3.99,
2.49, 4.99]
```



```
Grocery List:
Item            Price
--------------------
apple           $0.99
banana          $0.50
orange          $0.75
grapes          $2.99
tomato          $1.49
potato          $0.99
milk            $3.99
bread           $2.49
butter          $4.99
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

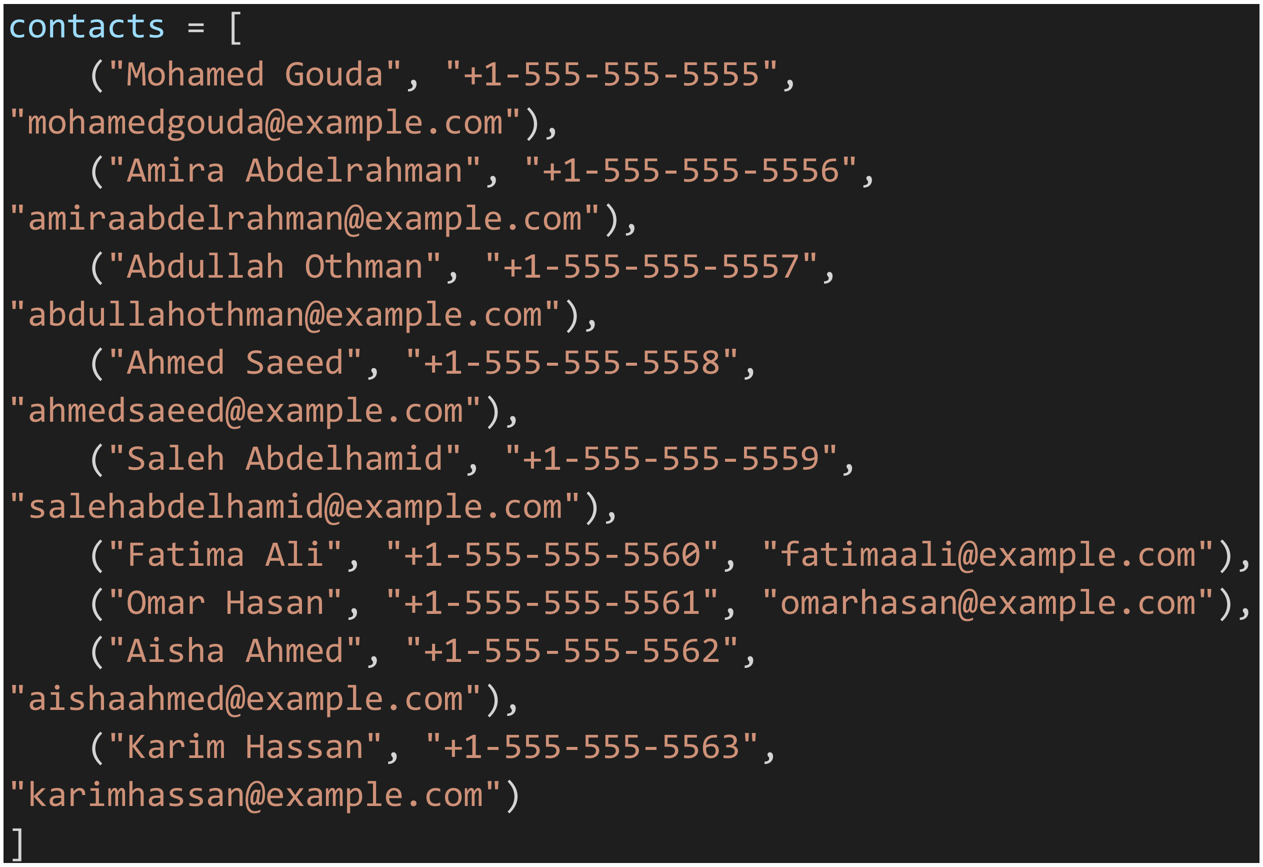
```
# 2. Grocery List program

# Create a list of tuples to store the grocery items and their
prices

# Display the grocery list

# Loop through the list of tuples and display the grocery items
and their prices
```



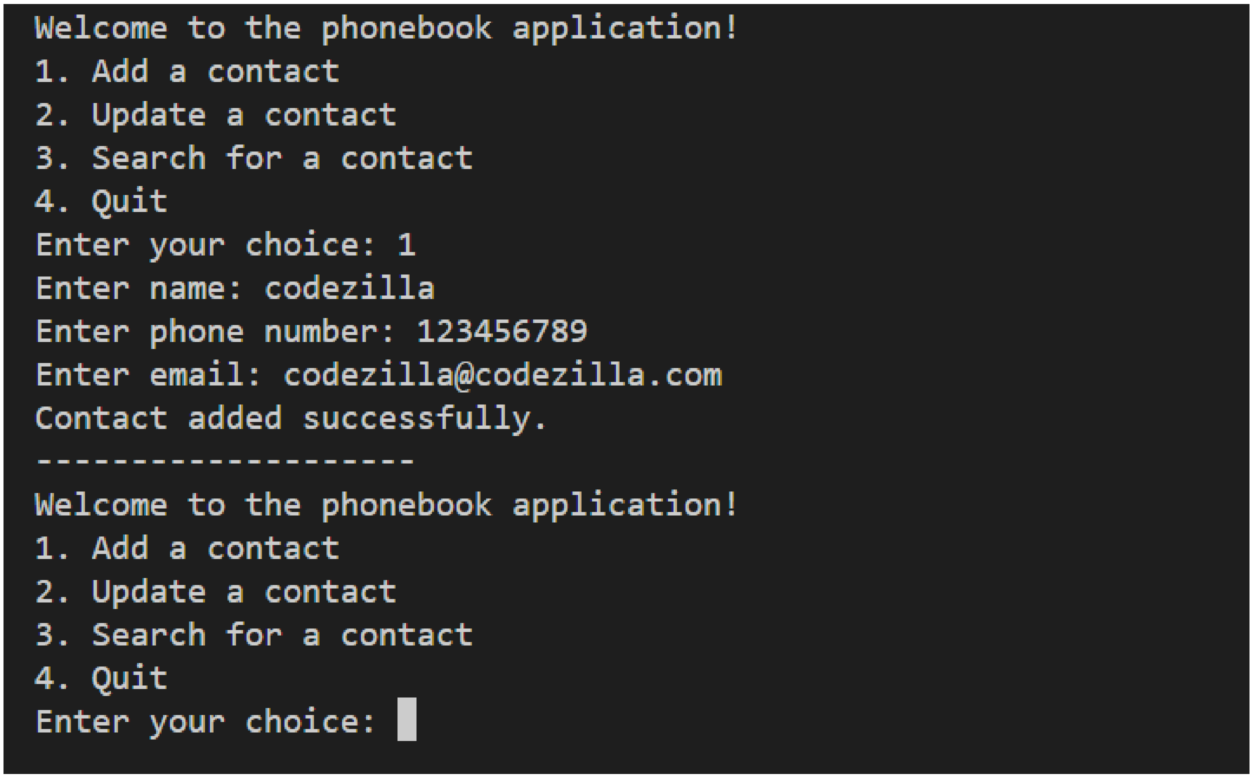
```python
# Create a list of tuples to store the grocery items and their prices
grocery_names = ["apple", "banana", "orange", "grapes", "tomato", "potato", "milk", "bread", "butter"]
grocery_prices = [0.99, 0.5, 0.75, 2.99, 1.49, 0.99, 3.99, 2.49, 4.99]

# Display the grocery list
print("Grocery List:")
print("Item\t\tPrice")
print("-------------------")

# Loop through the list of tuples and display the grocery items and their prices
for item, price in zip(grocery_names, grocery_prices):
    print(f"{item}\t\t${price:.2f}")
```

3.قم بعمل برنامج يتفاعل مع المستخدم بالشكل التالي باستخدام جهات الاتصال التالية

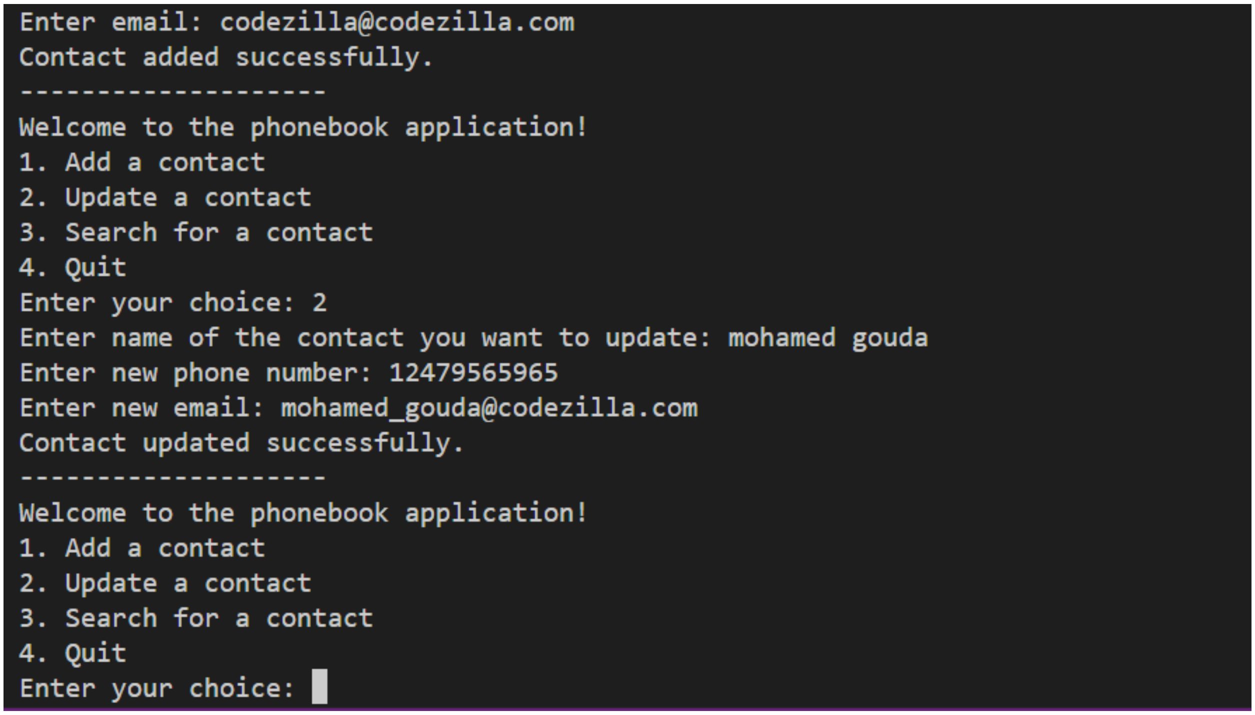
Make the following program.



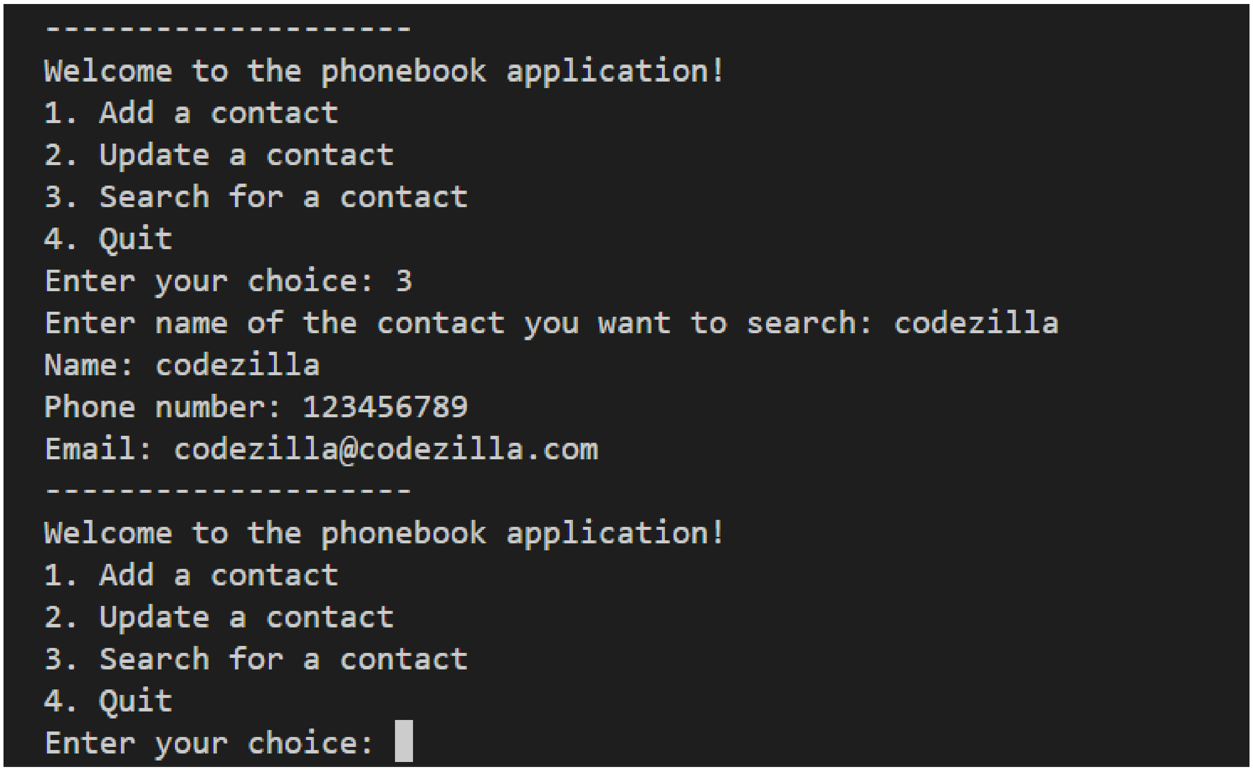
```
contacts = [
    ("Mohamed Gouda", "+1-555-555-5555",
"mohamedgouda@example.com"),
    ("Amira Abdelrahman", "+1-555-555-5556",
"amiraabdelrahman@example.com"),
    ("Abdullah Othman", "+1-555-555-5557",
"abdullahothman@example.com"),
    ("Ahmed Saeed", "+1-555-555-5558",
"ahmedsaeed@example.com"),
    ("Saleh Abdelhamid", "+1-555-555-5559",
"salehabdelhamid@example.com"),
    ("Fatima Ali", "+1-555-555-5560", "fatimaali@example.com"),
    ("Omar Hasan", "+1-555-555-5561", "omarhasan@example.com"),
    ("Aisha Ahmed", "+1-555-555-5562",
"aishaahmed@example.com"),
    ("Karim Hassan", "+1-555-555-5563",
"karimhassan@example.com")
]
```



```
Welcome to the phonebook application!
1. Add a contact
2. Update a contact
3. Search for a contact
4. Quit
Enter your choice: 1
Enter name: codezilla
Enter phone number: 123456789
Enter email: codezilla@codezilla.com
Contact added successfully.
--------------------
Welcome to the phonebook application!
1. Add a contact
2. Update a contact
3. Search for a contact
4. Quit
Enter your choice: 
```



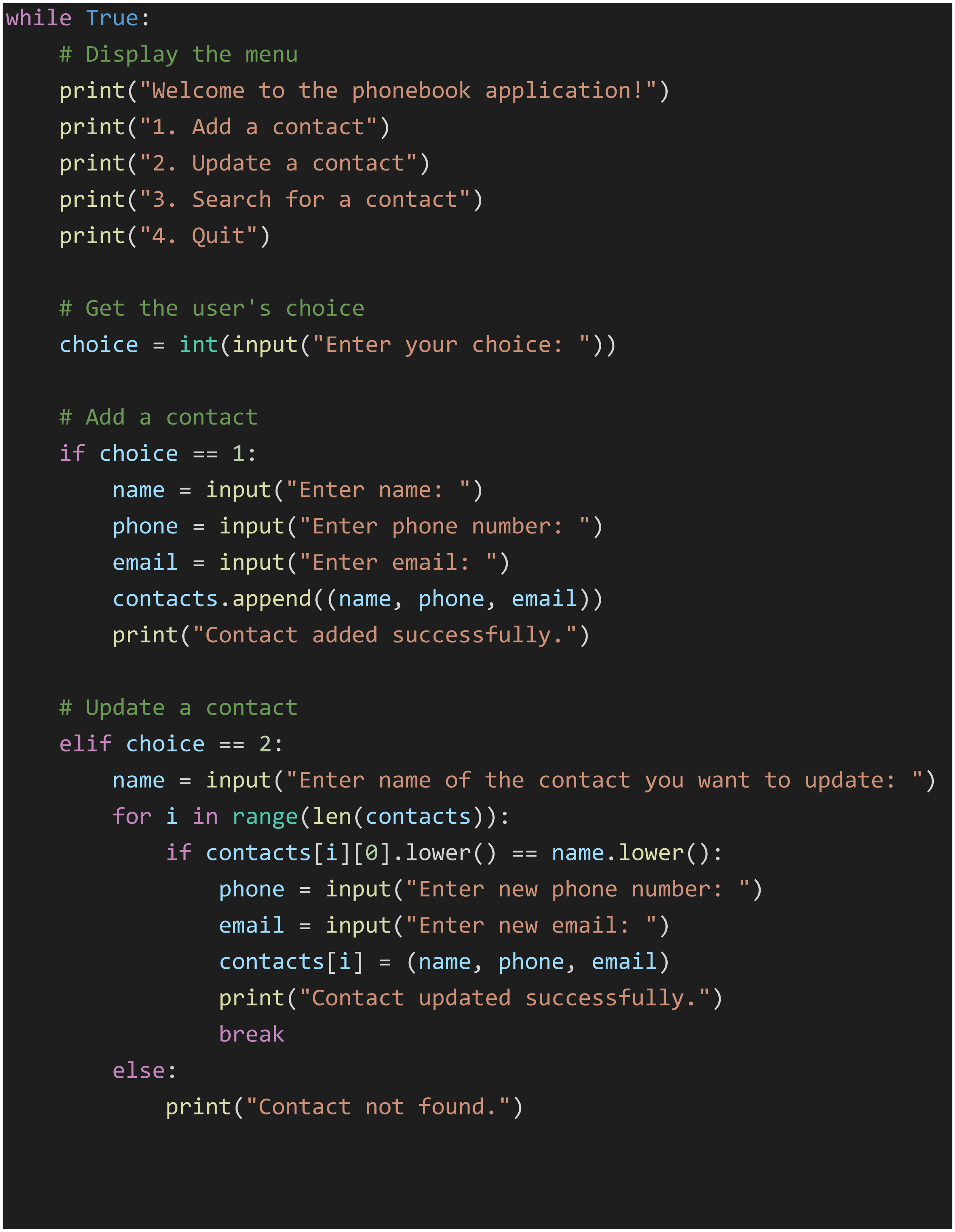
```
Enter email: codezilla@codezilla.com
Contact added successfully.
--------------------
Welcome to the phonebook application!
1. Add a contact
2. Update a contact
3. Search for a contact
4. Quit
Enter your choice: 2
Enter name of the contact you want to update: mohamed gouda
Enter new phone number: 12479565965
Enter new email: mohamed_gouda@codezilla.com
Contact updated successfully.
--------------------
Welcome to the phonebook application!
1. Add a contact
2. Update a contact
3. Search for a contact
4. Quit
Enter your choice: 
```



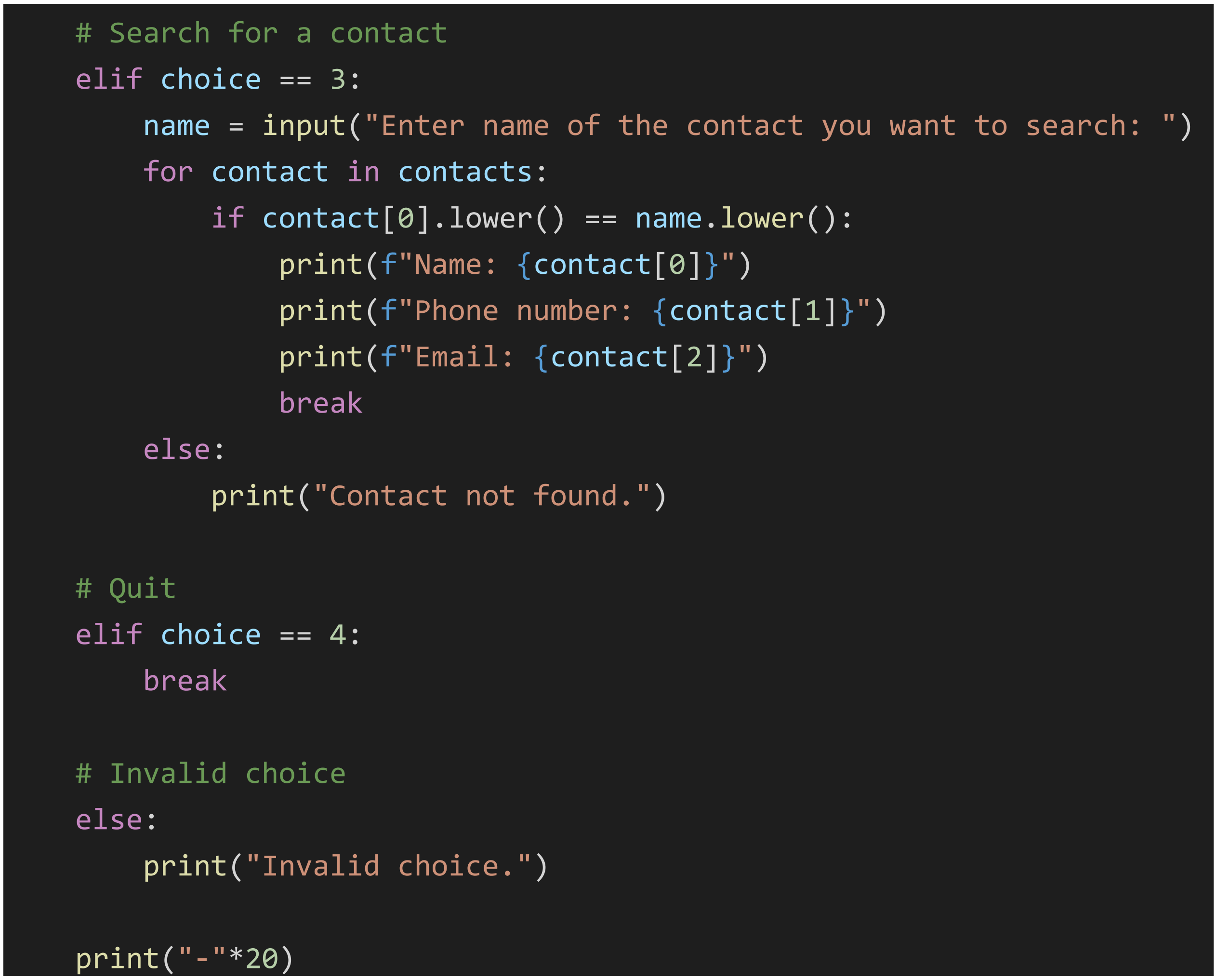
```
--------------------
Welcome to the phonebook application!
1. Add a contact
2. Update a contact
3. Search for a contact
4. Quit
Enter your choice: 3
Enter name of the contact you want to search: codezilla
Name: codezilla
Phone number: 123456789
Email: codezilla@codezilla.com
--------------------
Welcome to the phonebook application!
1. Add a contact
2. Update a contact
3. Search for a contact
4. Quit
Enter your choice: 
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية

```
# 3. Contact list program

# Display the menu

# Get the user's choice

# Add a contact

# Update a contact

# Search for a contact

# Quit

# Invalid choice
```



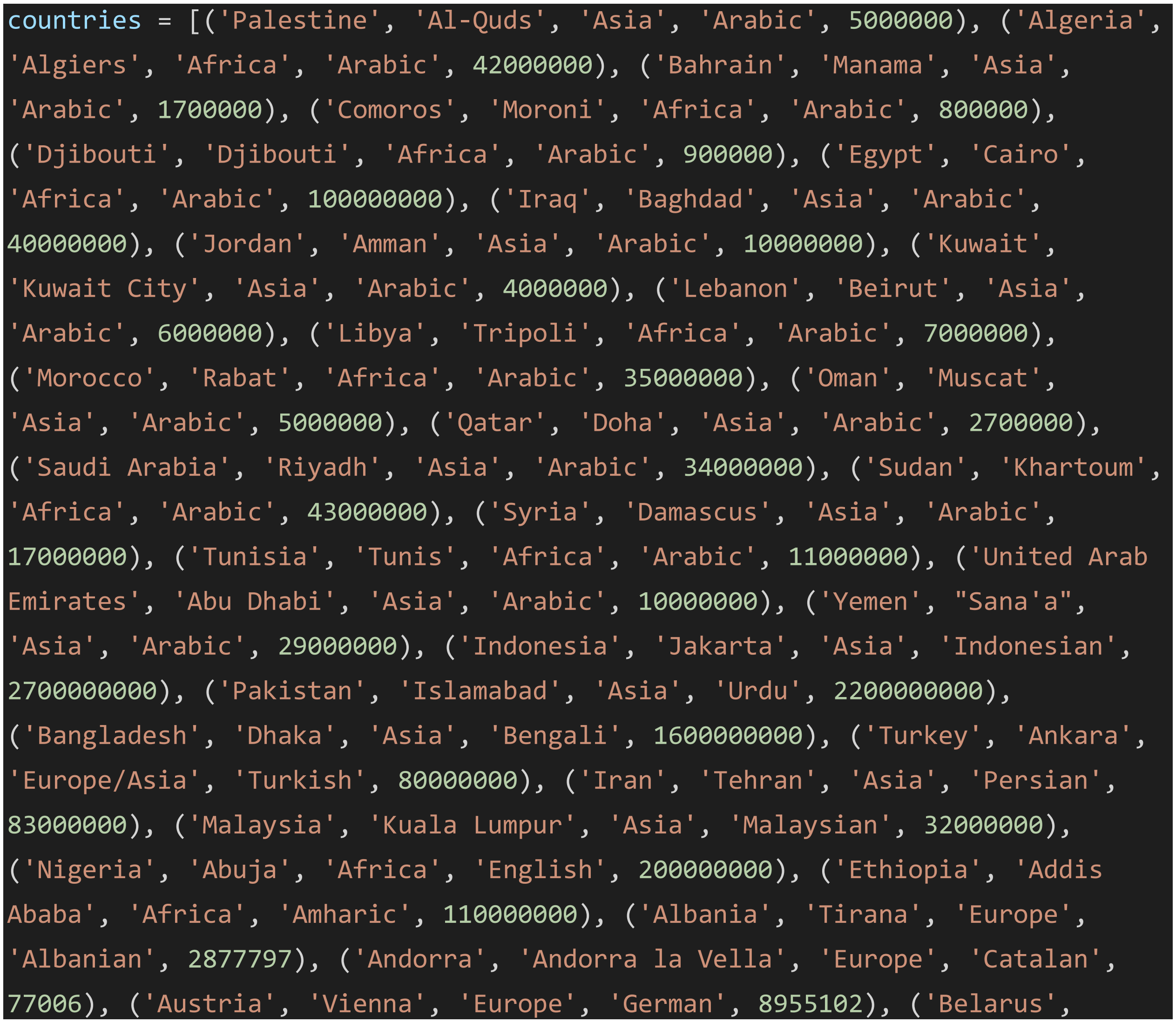
```
while True:
    # Display the menu
    print("Welcome to the phonebook application!")
    print("1. Add a contact")
    print("2. Update a contact")
    print("3. Search for a contact")
    print("4. Quit")

    # Get the user's choice
    choice = int(input("Enter your choice: "))

    # Add a contact
    if choice == 1:
        name = input("Enter name: ")
        phone = input("Enter phone number: ")
        email = input("Enter email: ")
        contacts.append((name, phone, email))
        print("Contact added successfully.")

    # Update a contact
    elif choice == 2:
        name = input("Enter name of the contact you want to update: ")
        for i in range(len(contacts)):
            if contacts[i][0].lower() == name.lower():
                phone = input("Enter new phone number: ")
                email = input("Enter new email: ")
                contacts[i] = (name, phone, email)
                print("Contact updated successfully.")
                break
        else:
            print("Contact not found.")
```



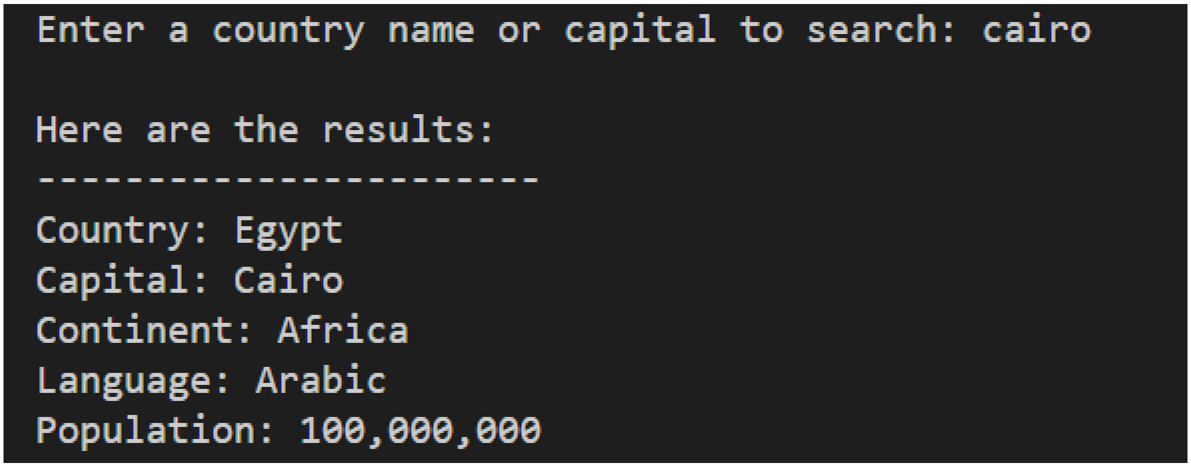
```
    # Search for a contact
    elif choice == 3:
        name = input("Enter name of the contact you want to search: ")
        for contact in contacts:
            if contact[0].lower() == name.lower():
                print(f"Name: {contact[0]}")
                print(f"Phone number: {contact[1]}")
                print(f"Email: {contact[2]}")
                break
        else:
            print("Contact not found.")

    # Quit
    elif choice == 4:
        break

    # Invalid choice
    else:
        print("Invalid choice.")

    print("-"*20)
```

4.قم بعمل برنامج يتيح للمستخدم البحث عن دولة ما أو عاصمة
داخل الدول التالية ثم يقوم بطباعة البيانات التالية

Make a program that allow the user to enter a country name and capital to search for then show the following information

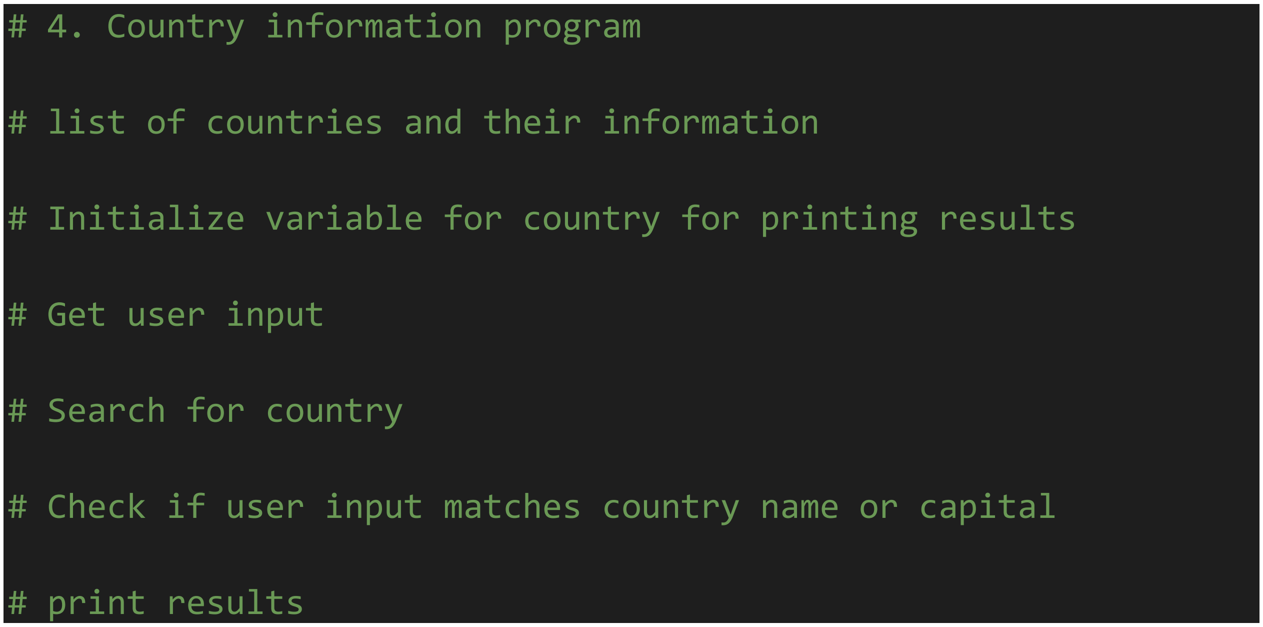


```
countries = [('Palestine', 'Al-Quds', 'Asia', 'Arabic', 5000000), ('Algeria',
'Algiers', 'Africa', 'Arabic', 42000000), ('Bahrain', 'Manama', 'Asia',
'Arabic', 1700000), ('Comoros', 'Moroni', 'Africa', 'Arabic', 800000),
('Djibouti', 'Djibouti', 'Africa', 'Arabic', 900000), ('Egypt', 'Cairo',
'Africa', 'Arabic', 100000000), ('Iraq', 'Baghdad', 'Asia', 'Arabic',
40000000), ('Jordan', 'Amman', 'Asia', 'Arabic', 10000000), ('Kuwait',
'Kuwait City', 'Asia', 'Arabic', 4000000), ('Lebanon', 'Beirut', 'Asia',
'Arabic', 6000000), ('Libya', 'Tripoli', 'Africa', 'Arabic', 7000000),
('Morocco', 'Rabat', 'Africa', 'Arabic', 35000000), ('Oman', 'Muscat',
'Asia', 'Arabic', 5000000), ('Qatar', 'Doha', 'Asia', 'Arabic', 2700000),
('Saudi Arabia', 'Riyadh', 'Asia', 'Arabic', 34000000), ('Sudan', 'Khartoum',
'Africa', 'Arabic', 43000000), ('Syria', 'Damascus', 'Asia', 'Arabic',
17000000), ('Tunisia', 'Tunis', 'Africa', 'Arabic', 11000000), ('United Arab
Emirates', 'Abu Dhabi', 'Asia', 'Arabic', 10000000), ('Yemen', "Sana'a",
'Asia', 'Arabic', 29000000), ('Indonesia', 'Jakarta', 'Asia', 'Indonesian',
2700000000), ('Pakistan', 'Islamabad', 'Asia', 'Urdu', 2200000000),
('Bangladesh', 'Dhaka', 'Asia', 'Bengali', 1600000000), ('Turkey', 'Ankara',
'Europe/Asia', 'Turkish', 80000000), ('Iran', 'Tehran', 'Asia', 'Persian',
83000000), ('Malaysia', 'Kuala Lumpur', 'Asia', 'Malaysian', 32000000),
('Nigeria', 'Abuja', 'Africa', 'English', 200000000), ('Ethiopia', 'Addis
Ababa', 'Africa', 'Amharic', 110000000), ('Albania', 'Tirana', 'Europe',
'Albanian', 2877797), ('Andorra', 'Andorra la Vella', 'Europe', 'Catalan',
77006), ('Austria', 'Vienna', 'Europe', 'German', 8955102), ('Belarus',
'Minsk', 'Europe', 'Belarusian', 9449323), ('Belgium', 'Brussels', 'Europe',
'Dutch, French, German', 11519193), ('Bosnia and Herzegovina', 'Sarajevo',
'Europe', 'Bosnian, Serbian, Croatian', 3280819), ('Bulgaria', 'Sofia',
'Europe', 'Bulgarian', 6948445), ('Croatia', 'Zagreb', 'Europe', 'Croatian',
4105267), ('Cyprus', 'Nicosia', 'Europe', 'Greek, Turkish', 1189265), ('Czech
Republic', 'Prague', 'Europe', 'Czech', 10618303), ('Denmark', 'Copenhagen',
'Europe', 'Danish', 5792202), ('Estonia', 'Tallinn', 'Europe', 'Estonian',
1324333), ('Finland', 'Helsinki', 'Europe', 'Finnish, Swedish', 5540720),
('France', 'Paris', 'Europe', 'French', 67102172), ('Germany', 'Berlin',
'Europe', 'German', 82979224), ('Greece', 'Athens', 'Europe', 'Greek',
10423054), ('Hungary', 'Budapest', 'Europe', 'Hungarian', 9769526),
('Iceland', 'Reykjavik', 'Europe', 'Icelandic', 341243), ('Ireland',
'Dublin', 'Europe', 'English, Irish', 4761657), ('Italy', 'Rome', 'Europe',
'Italian', 60461826), ('Kosovo', 'Pristina', 'Europe', 'Albanian, Serbian',
1800000), ('Latvia', 'Riga', 'Europe', 'Latvian', 1886198), ('Liechtenstein',
'Vaduz', 'Europe', 'German', 38375), ('Lithuania', 'Vilnius', 'Europe',
'Lithuanian', 2722289), ('Luxembourg', 'Luxembourg', 'Europe',
'Luxembourgish, French, German', 625978), ('Malta', 'Valletta', 'Europe',
'Maltese, English', 514000), ('Moldova', 'Chisinau', 'Europe', 'Romanian',
2883300), ('Monaco', 'Monaco', 'Europe', 'French', 39242), ('Montenegro',
'Podgorica', 'Europe', 'Montenegrin', 621810), ('Netherlands', 'Amsterdam',
'Europe', 'Dutch', 17231017), ('North Macedonia', 'Skopje', 'Europe',
'Macedonian', 2083374), ('Canada', 'Ottawa', 'North America', 'English,
French', 38550000), ('United States', 'Washington, D.C.', 'North America',
'English', 329000000), ('Mexico', 'Mexico City', 'North America', 'Spanish',
130750000), ('Brazil', 'Brasília', 'South America', 'Portuguese', 211200000),
('Argentina', 'Buenos Aires', 'South America', 'Spanish', 44480000),
('Colombia', 'Bogotá', 'South America', 'Spanish', 50370000), ('Peru',
'Lima', 'South America', 'Spanish', 32160000), ('Venezuela', 'Caracas',
'South America', 'Spanish', 28200000)]
```

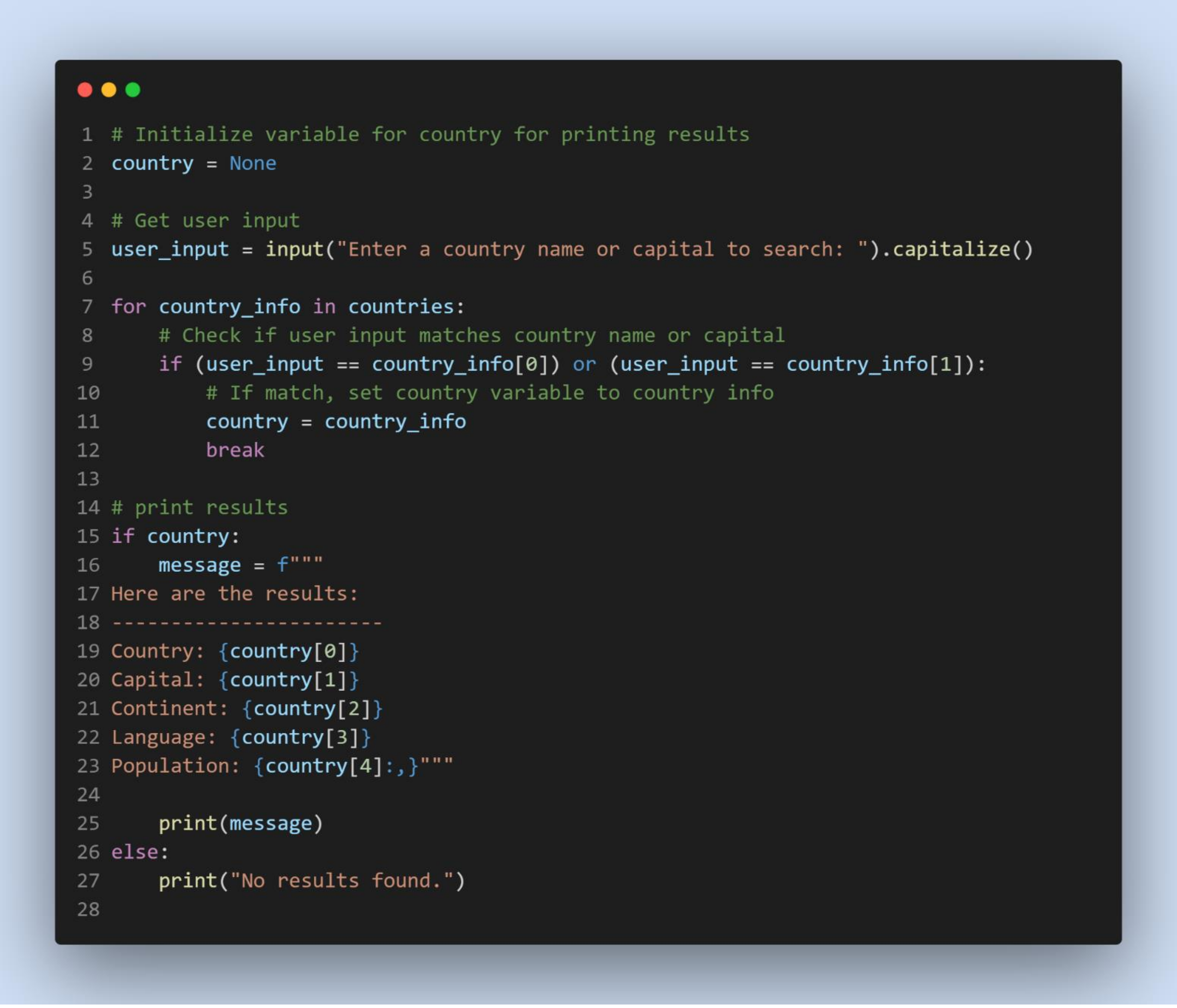


```
Enter a country name or capital to search: cairo

Here are the results:
-----------------------
Country: Egypt
Capital: Cairo
Continent: Africa
Language: Arabic
Population: 100,000,000
```

إذا لم توفق للوصول للحل يمكنك السعي مرة أخرى بمساعدة الخطوات التالية



```
# 4. Country information program

# list of countries and their information

# Initialize variable for country for printing results

# Get user input

# Search for country

# Check if user input matches country name or capital

# print results
```



```python
# Initialize variable for country for printing results
country = None

# Get user input
user_input = input("Enter a country name or capital to search: ").capitalize()

for country_info in countries:
    # Check if user input matches country name or capital
    if (user_input == country_info[0]) or (user_input == country_info[1]):
        # If match, set country variable to country info
        country = country_info
        break

# print results
if country:
    message = f"""
Here are the results:
-----------------------
Country: {country[0]}
Capital: {country[1]}
Continent: {country[2]}
Language: {country[3]}
Population: {country[4]:,}"""

    print(message)
else:
    print("No results found.")

```