

|  |  |
| --- | --- |
| **NAME :** | **TANISHA SOMA** |
| **ROLL NO. :** | **2021a1r029** |
| **BRANCH :** | **CSE** |
| **SECTION :** | **A1** |
| **SEMESTER :** | **5th** |
| **COURSE :** | **Python Programming** |
| **COURSE CODE :** | **COM-501** |

**ASSIGNMENT 1 1)**Write a complete COUNTRY MANAGEMENT APP. Your code should store COUNTRY CODE, COUNTRY NAME, CAPITAL CITY AND POPULATION as key-value pair in a dictionary and allow perform following operations on the dictionary:

View

Add

Delete

Ans:

class CountryManagementApp:

    def \_\_init\_\_(self):

        # Initialize the dictionary with provided key-value pairs

        self.country\_data = {

            'IN': {'name': 'India', 'capital': 'Delhi', 'population': 1320000000},

            'US': {'name': 'America', 'capital': 'Washington', 'population': 320000000},

            'AU': {'name': 'Australia', 'capital': 'Canberra', 'population': 24000000},

            'CA': {'name': 'Canada', 'capital': 'Ottawa', 'population': 940000}

        }

    def view\_countries(self):

        print("Country Codes:", end=" ")

        for code in self.country\_data.keys():

            print(code, end=" ")

        print()

        code = input("Enter country code: ").upper()

        if code in self.country\_data:

            country = self.country\_data[code]

            print(f"Country name is: {country['name']}")

            print(f"Country capital is: {country['capital']}")

            print(f"Country population is: {country['population']}")

        else:

            print("Invalid country code.")

    def add\_country(self, code, name, capital, population):

        # Check if the country code already exists

        if code in self.country\_data:

            print("Country code already exists. Use 'Update' option to modify.")

        else:

            # Add the new entry to the dictionary

            self.country\_data[code] = {'name': name, 'capital': capital, 'population': population}

            print(f"Country {name} added successfully.")

    def delete\_country(self, code):

        # Check if the country code exists

        if code in self.country\_data:

            # Remove the entry from the dictionary

            del self.country\_data[code]

            print(f"Country with code {code} deleted successfully.")

        else:

            print("Country code not found. Unable to delete.")

if \_\_name\_\_ == "\_\_main\_\_":

    # Create an instance of the CountryManagementApp

    country\_app = CountryManagementApp()

    while True:

        print("\nSELECT AN OPTION:")

        print("view: View country names")

        print("add: Add a country")

        print("delete: Delete a country")

        print("exit: Exit the program")

        choice = input("Enter your choice (view/add/delete/exit): ")

        if choice == 'view':

            country\_app.view\_countries()

        elif choice == 'add':

            code = input("Enter Country Code: ").upper()

            name = input("Enter Country Name: ")

            capital = input("Enter Capital City: ")

            population = int(input("Enter Population: "))

            country\_app.add\_country(code, name, capital, population)

        elif choice == 'delete':

            code = input("Enter Country Code to delete: ").upper()

            country\_app.delete\_country(code)

        elif choice == 'exit':

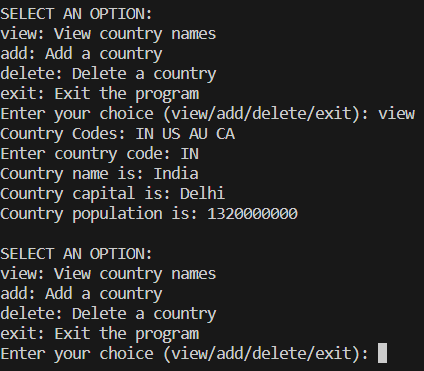
            print("Exiting Country Management App. Goodbye!")

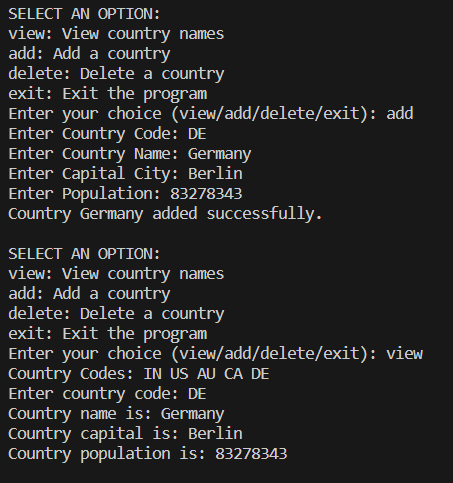
            break

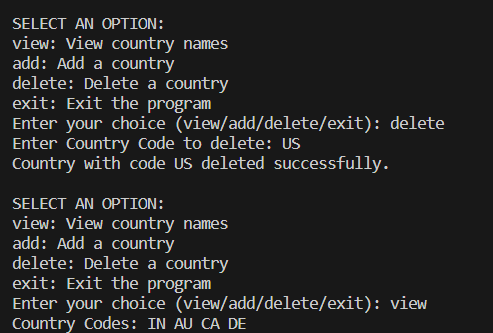
        else:

            print("Invalid choice. Please enter a valid option.")

**Output:**







1. Write a program to simulate user registration process.

Your code should do the following:

1. It should first ask the user to input his full name. If he doesn’t enter his full name then program should display the error message and again ask the user to enter full name. Repeat the process until the user types his full name. [ full name means a string with at least 2 words separated with a space]

2. Then it should ask the use to input his password. The rules for password are:

1.It should contain at least 8 characters

2.It should contain at least 1 digit and 1 upper case letter Repeat the process until the user correctly types his Password.

Finally, display the user’s first name with a THANK YOU message. Create separate functions for accepting full-name, password and returning first name.

Ans:

def get\_full\_name():

    while True:

        full\_name = input("Enter your full name: ")

        if ' ' in full\_name:

            return full\_name

        else:

            print("Please enter your full name (at least two words separated by a space).")

def validate\_password(password):

    has\_digit = any(char.isdigit() for char in password)

    has\_upper = any(char.isupper() for char in password)

    return len(password) >= 8 and has\_digit and has\_upper

def get\_password():

    while True:

        password = input("Enter your password: ")

        if validate\_password(password):

            return password

        else:

            print("Password should have at least 8 characters, 1 digit, and 1 uppercase letter.")

def get\_first\_name(full\_name):

    return full\_name.split()[0]

def main():

    full\_name = get\_full\_name()

    password = get\_password()

    first\_name = get\_first\_name(full\_name)

    print(f"THANK YOU, {first\_name}!")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

