EXPERIMENT-6

STATE TO CAPITAL FINDER

1.Research

During my research on these two programs—one based on a telephone directory and the other based on Indian states and their capitals—I found that both topics are strongly connected in terms of logic, structure, and the way data is handled. At first, they look completely different: one deals with phone numbers and locations, while the other deals with states and capitals. But when I studied them closely, I realized both programs depend on the same fundamental concept of **mapping one piece of information to another**. In the telephone directory, a phone number is linked to its location using an array. Similarly, in the state–capital program, each state is linked to its capital using parallel arrays. In both cases, the user gives an input, and the program must **search** through stored records to find the correct match. Both programs rely on **linear search**, **string or number comparison**, and the idea that the correct answer exists at the **same index** in another array. This made me understand that programming is not about memorizing topics separately, but about recognizing patterns and applying the same logic to different types of data. Whether it is finding a capital from a state or a location from a phone number, both programs teach the same essential skill: how computers look up and retrieve information quickly. This connection helped me see how search operations work in real apps like contacts, Google Maps, or even phone books, which internally follow the same principle of matching user input with stored data. Because of this, I now understand that both programs are basically **different examples of the same underlying concept—data lookup using arrays and search techniques.**

References

1.https://en.wikipedia.org/wiki/List_of_state_and_union_territory_capitals_in_India

2. https://chahalacademy.com/indian-states-and-their-capitals

3. https://www.geeksforgeeks.org/social-science/states-and-capitals-in-india/

4.https://timesofindia.indiatimes.com/blogs/issues-from-the-ground-and-experience/geographical-centrality-of-capital-cities-haryana-and-punjab-are-exceptions/

5. https://en.wikipedia.org/wiki/Telephone_directory

2.Analysis

After studying both the telephone directory program and the state–capital program, I analyzed that the two topics are closely connected because they follow the same structure of data handling and searching. In both programs, two related pieces of information are stored side by side in separate arrays—phone numbers with locations in one case, and states with their capitals in the other. When a user enters a value, the program compares it with the stored list one element at a time using a loop. This process, known as linear searching, is the central logic behind both tasks. Although the data looks different, the working method is exactly the same: **take input → compare with stored records → if match found → show linked information**. This shows that both programs are examples of basic information-retrieval systems, where the computer fetches the right output based on the index of the matching input. Through this analysis, I understood that learning one such program automatically helps in understanding the other, because the underlying logic—arrays, loops, and matching—is identical. The difference is only in the type of data, not in the technique. This makes both topics strongly related from a programming and problem-solving perspective.

3.Ideate

While ideating and trying to understand both the telephone directory program and the state–capital program, I realized that the main idea behind them is the same: connecting one piece of information with another using a simple data structure. For the telephone directory, the idea was to link a phone number with a location. For the state–capital program, the idea was to link each state with its capital. This led me to think in terms of **paired data**, where two arrays can store related information in the same index. I also noticed that both programs require the user to enter a key—either a phone number or a state name—and the system's job is to match that key and return the correct value. This similarity gave me the idea of using **linear search** as the common method for both tasks. Instead of treating them as two separate problems, I started seeing them as two different applications of the same concept: searching and retrieving data. This ideation process helped me understand how many real-life tools like contact apps, dictionaries, and map applications work internally. Both programs are simply small models of bigger systems that use the same idea of "input to output mapping." This made the concept clear and easy to build upon.

4.BUILD

```
#include <stdio.h>

#include <string.h>
```

```
struct State {
    char name[40];
    char capital[40];
};


int main(void) {
    struct State s[] = {
        {"Andhra Pradesh", "Amaravati"},
        {"Arunachal Pradesh", "Itanagar"},
        {"Assam", "Dispur"},
        {"Bihar", "Patna"},
        {"Chhattisgarh", "Raipur"},
        {"Goa", "Panaji"},
        {"Gujarat", "Gandhinagar"},
        {"Haryana", "Chandigarh"},
        {"Himachal Pradesh", "Shimla"},
        {"Jharkhand", "Ranchi"},
        {"Karnataka", "Bengaluru"},
        {"Kerala", "Thiruvananthapuram"},
        {"Madhya Pradesh", "Bhopal"},

        {"Maharashtra", "Mumbai"},

        {"Manipur", "Imphal"},

        {"Meghalaya", "Shillong"},

        {"Mizoram", "Aizawl"},

        {"Nagaland", "Kohima"},

        {"Odisha", "Bhubaneswar"},

        {"Punjab", "Chandigarh"},

        {"Rajasthan", "Jaipur"},

        {"Sikkim", "Gangtok"},

        {"Tamil Nadu", "Chennai"},

        {"Telangana", "Hyderabad"},

        {"Tripura", "Agartala"},

        {"Uttar Pradesh", "Lucknow"},

        {"Uttarakhand", "Dehradun"},

        {"West Bengal", "Kolkata"}

    };


    int total = sizeof s / sizeof s[0];
```

```
char input[40];

int found = 0;


printf("Enter state name: ");

if (fgets(input, sizeof input, stdin) == NULL) {

    printf("Input error.\n");

    return 1;

}

for (int i = 0; input[i] != '\0'; i++) {

    if (input[i] == '\n') { input[i] = '\0'; break; }

}


for (int i = 0; i < total; i++) {

    if (strcmp(input, s[i].name) == 0) {

        printf("Capital of %s is %s\n", s[i].name, s[i].capital);

        found = 1;

        break;

    }

}
```

```
    if (!found) puts("Please enter a valid state name.");
    return 0;
}
```

5.TESTING

Case 1;

 (After putting any random letters)

```
Enter state name: yhfj
State not found in database.
```

Case 2;

(After entering any digit)

```
Enter state name: 67
Please enter a valid state name.
```

Case 3;

(After entering any capital name)

```
Enter state name: Mumbai
Please enter a valid state name.
```

Case 4;

(After entering any special symbol)

```
Enter state name: @maharshtra

Please enter a valid state name.
```

6.FINAL OUTPUT

```
1]      Enter state name: Manipur

        Capital of Manipur is Imphal



2]      Enter state name: Rajasthan

       Capital of Rajasthan is Jaipur
```

7. IMPLEMENTATION

The project is published on github for easy access:

8. CONCLUSION

This program uses structures and string handling to display the capital of the entered Indian state. It also validates user input to prevent numbers and ensures accurate and meaningful output.

THANK YOU!!