1. Store Student Info

Q: Create a dictionary to store a student's name, age, class, and roll number. Then display the information in a readable format.

Hint: Use a dictionary like:

```
student = {"name": "Ram", "age": 15, "class": "10", "roll": 23}
```

Use for loop or print() statements to display data.

---

2. Phone Book

Q: Create a simple phone book dictionary where names are keys and phone numbers are values. Let the user input 3 contacts and display them.

Hint:

- Use a for loop and dict[key] = value
- Example: contacts["Sita"] = "9876543210"

---

3. Marks Calculator

Q: Ask the user to enter subject names and corresponding marks (for 5 subjects). Store them in a dictionary and calculate total and average.

Hint:

- Use marks[subject] = mark
- Use sum(marks.values()) for total

---

4. Check if Key Exists

Q: Create a dictionary of 5 fruits with their prices. Ask the user to enter a fruit name and display its price if it exists.

Hint:

- Use if fruit in dict: to check if it exists.

---

5. Count Word Frequency in a Sentence

Q: Ask the user to enter a sentence. Count how many times each word occurs using a dictionary.

Hint:

- Use .split() to split words.
- Use a loop to add words to a dictionary with their count.

---

6. Update Dictionary

Q: Create a dictionary with some country–capital pairs. Ask user to update capital of a given country.

Hint:

- Use: dict["Nepal"] = "Kathmandu" to update value

---

7. Delete an Entry

Q: Let the user delete a student from a class record (dictionary of names and roll numbers).

Hint: Use del dict[key]

---

8. Merge Two Dictionaries

Q: Take two dictionaries and merge them into one.

Hint: Use:

```
dict3 = dict1.copy()

dict3.update(dict2)
```

---

9. Reverse Key and Value

Q: Ask user to enter a few English–Nepali word pairs in a dictionary. Then reverse the dictionary (Nepali–English).

Hint:
Use a loop:

```
reversed_dict = {v: k for k, v in original.items()}
```

---

10. Student Grades Dictionary

Q: Create a dictionary where student names are keys and their list of marks is the value. Calculate average marks for each student.

Hint:

```
data = {"Ram": [80, 85, 78], "Sita": [90, 92, 88]}
```

Menu-Driven Programs (do as much as you can)

(menu options, loops, and functions recommended)

---

1. Simple Calculator

Q: Build a menu-driven calculator that performs addition, subtraction, multiplication, and division based on user input.

Hint:

1. Add

2. Subtract

3. Multiply

4. Divide

5. Exit

Use if-elif, take two numbers, perform operation.

---

2. Student Record System

Q: Menu-driven program to:

1. Add student

2. Delete student

3. View all students

4. Exit

Hint: Use dictionary to store roll number and name.

---

3. Banking System

Q: Menu with options:

1. Check balance
2. Deposit money
3. Withdraw money
4. Exit

Hint:

- Store balance in a variable.
- Use loop + conditionals.

---

4. To-Do List

Q: Create a menu-based to-do list:

1. Add task
2. View all tasks
3. Remove task
4. Exit

Hint: Use a list to store tasks.

---

5. Library System

Q: Menu:

1. Add book
2. Issue book
3. Return book
4. Display books
5. Exit

Hint:

- Use dictionary to store book name → availability (True/False)