

## Exercises: functions

### Exe. #35:

Write a Python program that asks the user to input a word and a number. The program should then call a function that encodes the word using a Caesar cipher (i.e., shifting each letter by the number provided). The Caesar cipher should wrap around the alphabet (e.g., shifting “z” by 1 becomes “a”) using only lowercase letters.

code

Alternative solution (memorized alphabet)

code

### Exe. #36:

Write a Python program to manage a basic structure for storing and analyzing hospital patient data. The program should allow the user to perform several actions related to patient records.

- “**insert**” – Add a new health metric (blood\_pressure, glucose, ...) for a patient.
- “**average**” – Compute the average of a specific metric for a patient.
- “**delete**” – Remove a patient from the dataset.
- “**risk**” – Evaluate if a patient is at risk based on inserted metrics.
- “**summary**” – Print all stored data for a patient.

code

## Proposed Exercises

### Proposed #49:

Write a Python program to **check if a player has won** a game of Tic Tac Toe. A Tic Tac Toe board is represented as a **3×3 list of lists**, like this:

```
game = [
[1, 2, 0],
[2, 1, 0],
[2, 1, 1]
]
```

Each cell contains: - 0 -> empty square - 1 -> player 1's move - 2 -> player 2's move

Write a function that, given a board configuration, checks **if any player has won**, and **which player** it is. A player wins if they have **three of their tokens in a row** (vertical, horizontal, diagonal).

You can assume that there is **at most one winner** per board and that you do **not** need to validate how the moves were made.

code

### Proposed #50:

Write a program that asks the user to enter a sequence of temperatures in Celsius, separated by “,” and converts it to Fahrenheit equivalents using the formula:  $(^{\circ}\text{C} \times 9/5) + 32$ . Use a subprogram to perform the conversion.

code