

## VPJEX Test 3 Prep

### Question 1:

**Write a function** `numberGame(limit, divisor1, divisor2)` that performs the following:

- If any of the three arguments are **undefined**, prompt the user to enter values using `prompt()`.
- `limit` must be a **positive integer greater than 0**.
- `divisor1` and `divisor2` must be **integers greater than 1**.
- If **invalid values** are provided, display an `alert()` and **restart the function** (recursively).
- Loop from **1 to limit** and:
  - Print **"Alpha"** if the number is divisible by `divisor1`.
  - Print **"Beta"** if the number is divisible by `divisor2`.
  - Print **"AlphaBeta"** if the number is divisible by both.
  - Otherwise, print the number.
- Return the **count of numbers that printed "AlphaBeta"**.
- **Note:** Only the function should be submitted. Do not include code outside the function.

## Question 2:

### Student Age Tracker

Write a function named `ageTracker()` that prompts the user for student ages and returns basic statistics:

- Repeatedly prompt the user to enter an **age between 5 and 100** using `prompt()`.
- Stop when the user presses **Cancel**.
- Store only **valid numeric ages** in an array.
- After all input is done:
  - Print the **average, youngest, and oldest** ages using **template literals (String Interpolation)**.
  - Format the average to **two decimal places** using `.toFixed(2)`.
  - Return an **object** in the form: `{ avg, youngest, oldest, count }`.
- **Note:** Only the function should be submitted. Anything outside the function will not be marked.

### Question 3:

#### Simple List Manager

Build a small menu-driven console app named `listManager()` that allows users to manage a list of words interactively using only `prompt()` and `alert()`: **Note the list referred to here is an array.**

- Display the following menu inside a `while(true)` loop:
- Simple List Manager:
  - 1. Add word
  - 2. Remove last word
  - 3. Show list & word count
  - 4. Clear list
  - 5. Quit
- Implement each option:
  - **Add word:** Prompt the user for a word (non-empty string) and add it to the list.
  - **Remove last word:** Remove the last word from the list.
  - **Show list & word count:**
    - Display all words separated by commas.
    - Show the total word count using `alert()`.
  - **Clear list:** Empty the list completely.
  - **Quit:** Exit the loop and end the program.
- Store the list in a variable with **scope outside the loop** (module-level).