

Q1) Write a function that checks for lowercase characters.

- Prototype: `def islower(c):`
- Returns True if c is lowercase
- Returns False otherwise
- You are not allowed to import any module
- You are not allowed to use `str.upper()` and `str.isupper()`

Q2) Write a function that prints a string in uppercase followed by a new line.

- Prototype: `def uppercase(str):`
- You can only use no more than 2 print functions with string format
- You can only use one loop in your code
- You are not allowed to import any module
- You are not allowed to use `str.upper()` and `str.isupper()`

Q3) Write a function that prints the last digit of a number.

- Prototype: `def print_last_digit(number):`
- Returns the value of the last digit
- You are not allowed to import any module

Q4) Write a function that adds two integers and returns the result.

- Prototype: `def add(a, b):`
- Returns the value of $a + b$

Q5) Write a function that computes a to the power of b and return the value.

- Prototype: `def pow(a, b):`
- Returns the value of a^b
- You are not allowed to import any module

Q6) Write a Python function `sum_integers` that takes a variable number of integer arguments using `*args` and returns the sum of all the integers passed as arguments.

Q7) Write a Python function `merge_dicts_with_default` that takes an arbitrary number of dictionaries as arguments using `**kwargs`, merges them into a single dictionary, and returns the merged dictionary. If a key exists in multiple dictionaries, the value from the last dictionary should be used.

Q8) Write a Python function `greet` that takes a name as a required argument and an optional argument `salutation` with a default value of "Hello". The function should return a customized greeting message using the provided name and salutation.

Q9) Write a Python function `concatenate_strings` that takes a variable number of strings using `*args` and an optional separator argument `sep` with a default value of an empty string. The function should concatenate all the strings using the provided separator.

