HW3 CS103 Fall 2025. due: 10.12.2025

Grading & Deliverables

Grading: Each problem is worth 20 points. A problem is considered correct if all automated tests pass.

Deliverables: Submit `hw3.ipynb` and `independent_completion_form` for grading.

Submission Note: Ensure the script file is named `hw3.ipynb` to avoid penalties.

Rubric: Each problem: 20 points

Correct implementation: All tests pass

Naming conventions: Exact function and file naming required. Ensure to use proper variable names.

** Test your functions for edge cases. Functions must adhere to the exact names and argument structures provided to pass the automated tests.

Mandatory Functions

Mandatory Functions: Implement the following functions with the exact names and parameters specified. Call these functions as demonstrated below.

```
def myName():
    return "James Bond"
def myBlazerID():
    return "jbon12"

# Call these functions
print("My Name is =", myName(), " and my BlazerId is =",myBlazerID())
```

HW3 problems

Please use the same names for the functions and the input parameters.

`shortenStrings(lst, num)`

Write a function `shortenStrings(lst, num)` that takes a list of strings `lst` and an integer `num`. The function returns a new list containing only the first `num` characters of each string in `lst`. If a string's length is less than `num`, include the whole string.

Sample Inputs	**Sample Outputs**

`lst = ["Programming", "is", "fun"]`
`num = 4`
	`["Prog", "is", "fun"]`

`lst = ["Testing", "functions", "in", "Python"]`
`num = 5`
	`["Testi", "funct", "in", "Pytho"]`

`lst = ["A", "short", "test"]`
`num = 2`
	`["A", "sh", "te"]`

`extractUpper(s)`

Write a function `extractUpper(s)` that takes a string `s` and returns a new string containing only the uppercase characters from `s`.

Sample Inputs	**Sample Outputs**
`s = "The Quick Brown Fox"`	`"TQBF"`
`s = "Hello WORLD"`	`"HWORLD"`
`s = "abcDEFghiJKL"`	`"DEFJKL"`

indexOfVowels(s)

Write a function `indexOfVowels(s)` that takes a string `s` and returns a list of integers representing the positions (indices) of all the vowels (a, e, i, o, u) in the string.

Sample Inputs	**Sample Outputs**
`s = "Hello World"`	`[1, 4, 7]`
`s = "Data Science"`	`[1, 3, 7, 8, 11]`
`s = "Python"`	`[4]`

Problem 4: dataTypeChecker(x)

Write a function `dataTypeChecker(x)` that takes an argument `x` and performs the following actions based on the `type of x`. The function should return a different value based on the type of x:

If x is a **string**: return the **string in uppercase**.
If x is a **list**: return the **list sorted in ascending order**.
If x is a **float**: return the **float squared**.
For any other type, return the message: "**Unrecognized data type**" (*string type)

Sample Inputs	**Sample Outputs**	**Hints**
`x = “hello”`	`“HELLO”`	`x is a string`
`x = [3, 1, 2]`	`[1, 2, 3]`	`x is a list`
`x = 5.2`	`27.04`	`x is a float`
`x = 42`	`"Unrecognized data type"`	`x is an int`
`x = (3, 4)`	`"Unrecognized data type"`	`x is a tuple`

`sumOfDigits(n)`

Write a function `sumOfDigits(n)` that takes an integer `n` and returns the sum of its digits. Assume `n` is a positive integer.

Sample Input:	Expected Output:
n = 1234	10
n = 29000000	11
n = 10001	2