

CS103 – Fall 2025 – **Lab11**

General Instructions

1. In today's lab, you are required to complete all required exercises during class to receive attendance credit.
2. You are welcome to:
 - Work together with classmates.
 - Search online for help or documentation.
 - Ask the TA for guidance if you are stuck.
3. To receive credit, you must finish and show your solutions to the TA before leaving lab.
4. There are also extra (optional) exercises at the end for those who want to challenge themselves. These are not required for attendance credit but are recommended for practice.

Exercise Instructions

- Make a folder **Lab11** inside your **cs103fa25** folder.
- Create a new python file inside your Lab11 folder (lab11.py).

Required Exercises

EXERCISE 1:

Write a function “**twoDmatrix**” that takes two int “**n2**” and “**m**” and doesn’t return anything. The function prints a matrix such that $1 \leq n2 \leq 10$ and $0 \leq m \leq 99$ (you can assume these inputs). The function must print the numbers from 0 to **m**, printing **n** numbers per line. The numbers must be separated by two or three spaces. Hint: Use output formatting.

Sample Input:

n2 = 6

m = 21

EXERCISE 2:

Write a function **armstrongChecker** that takes an int "**a**" and returns a **Boolean Value** (True/False). The function checks whether a given number is an Armstrong number or not. An Armstrong number is a number such that the sum of its digits raised to the third power is equal to the number itself (Assume **a** is a three-digit number: 100-999).

Sample Input:

a = 371

Sample Output

True

Hints: $3^{**3} + 7^{**3} + 1^{**3} = 371$.

EXERCISE 3:

Write a function "**cEmail**" that takes a string "**s1**" and returns another string. Assuming that **s1** is an email address in the "username@companyname.com" format, please write function to print the company name of a given email address. Both usernames and company names are composed of letters only.

Sample Input

s1= "unan@uab.edu"

Sample Output

"uab"

Extra (Optional) Challenges

These are not required for credit but will help you practice and strengthen your problem-solving skills.

Challenge 1: “**basicStats**”

Write a function “**basicStats**” that takes a NumPy array “**npArray**” and returns a list. The list will contain three values: mean, standard deviation, and variance of the input array.

Sample Input:

```
npArray = [0,1,2,3,4,5]
```

Sample Output:

```
[2.5, 1.707825127659933, 2.9166666666666665]
```

hints:

- mean = 2.5
- std = 1.707825127659933
- var = 2.9166666666666665

Challenge 2: **firstLastOccurrence(s)**

Write a program to find the first and the last occurrence of the letter 'x' in any string. Your function should return the sum of these two index values

Sample Input

```
s = "exit execution"
```

Sample Output

```
7  Hints: 1+6=7
```

Challenge 3: **palCreator**

Write a function “**palCreator**” that takes an int “**n**”. The function is to check whether **n** is a palindrome or not. If it is a palindromic number, it should print all the other palindromic numbers that can be formed by using the same digits.

* A palindromic number is a 'symmetrical' number like 16461, that remains the same when its digits are reversed.

Sample Input

4325234

Sample Output

4235324 3425243 3245423 2345432 2435342

Challenge 4: "Sudoku Validator"

Write a function `validateSudoku(grid)` that takes a 9×9 nested list representing a Sudoku board and checks if it is a valid Sudoku solution. A valid solution must satisfy the following conditions:

1. Each row contains the numbers 1–9 exactly once.
2. Each column contains the numbers 1–9 exactly once.
3. Each 3×3 subgrid contains the numbers 1–9 exactly once.

The function should return `True` if the board is valid, otherwise `False`.

Sample Input:

```
grid = [  
    [5,3,4,6,7,8,9,1,2],  
    [6,7,2,1,9,5,3,4,8],  
    [1,9,8,3,4,2,5,6,7],  
    [8,5,9,7,6,1,4,2,3],  
    [4,2,6,8,5,3,7,9,1],  
    [7,1,3,9,2,4,8,5,6],  
    [9,6,1,5,3,7,2,8,4],  
    [2,8,7,4,1,9,6,3,5],  
    [3,4,5,2,8,6,1,7,9]  
]
```

Expected Output: True

To get attendance credit, finish Exercises 1–3.

If you finish early, try the optional challenges!