MADRAS SCHOOL OF ECONOMICS

UNDERGRADUATE PROGRAMME IN ECONOMICS (HONOURS) [2023-26]

SEMESTER 6 [JANUARY – MAY, 2025]

REGULAR EXAMINATION, MAY 2025

Course Name: Introduction to MATLAB/R/Python Course Code: AE04

Duration: 2 Hours

Maximum Marks: 60

Instructions: The exam has two rounds.

In the first round, you will answer the exam on paper without access to a computer. **Once you turn in the written sheet the second round starts.** In this round, you can receive your computer and log in to classtime in front of an invigilator and answer the coding questions. You will not be given the written script back once its submitted.

All 1 markers are to be written in paper.

Part A: Answer all questions (1 mark x 10 questions = 10 marks)

- 1. (MATLAB) What will be printed?
 - x = 1:5; y = x.^2; disp(y(3))
- 2. (Python) What is the output?

lst = [2, 4, 6, 8]
print(lst[2])

3. (MATLAB) What is the output of the following code?

```
A = [1 2; 3 4];
size(A)
```

4. (Python) What is printed?

def add(a, b):
 return a + b
print(add(3, 5))

- 5. (MATLAB) What are the values in x?
 - x = linspace(0,1,5)
- 6. (Python) What is the output?

```
x = [i for i in range(5) if i%2==0]
print(x)
```

7. (MATLAB) What value is printed?

```
x = 0;
for i = 1:3
     x = x + i;
end
disp(x)
```

8. (Python) What is the output of the loop?

```
x = 0
for i in range(1,4):
    x += i
print(x)
```

9. (MATLAB) Write down the correct function call to compute the square of 4 for this definition.

```
function y = square(n)
    y = n^2;
end
```

10. (Python) What is the output?

```
lst = []
for i in range(3):
    lst.append(i*2)
print(lst)
```

Part B: Answer any five. (5 mark x 10 questions = 50 marks)

- 11. (Written) Answer the following essay questions:
 - (a) (6 marks) Compare the similarities and differences between MATLAB and Python.
 - (b) (4 marks) What is a class in Python ? Explain its structure.
- 12. (Written) Answer the following questions:
 - (a) (5 marks) The following MATLAB code adds numbers divisible by 3 to s, and numbers divisible by 7 to s1, skipping over numbers divisible by 3 after adding.

Fill in the blanks and state the final values of ${\tt s}$ and ${\tt s1}.$

```
L = [10, 18, 25, 14, 28, 9, 6];
s = 8;
s1 = 2;
for i = L
    if mod(i, ___) == 0
        s = s + i;
        continue;
    end
    if mod(i, ___) == 0
        s1 = s1 + i;
    end
end
```

What are the final values of s and s1?

(b) (5 marks) The following Python program prints a pattern based on the number entered by the user.

```
rounds = int(input("How many levels to update?"))
```

```
def update_pattern(rounds):
    for step in range(rounds):
        for pos in range(step + 1):
            print("5", end=" ")
        print()
```

update_pattern(rounds)

- (a) What is the output if the user inputs 5? (3 marks)
- (c) What role does step + 1 play in the pattern? (2 marks)

13. (Classtime) Word Histogram

Write a function called word_histogram(text) that takes a string and returns a dictionary where:

- Keys are lowercase words (case-insensitive),
- Values are the number of times each word appears.

Ignore punctuation and assume words are separated by spaces.

Function Signature:

```
def word_histogram(text):
    """
    Takes a string and returns a dictionary mapping words (lowercased, no punctuation)
    to their frequency counts.
    """
    pass
```

Example:

```
>>> word_histogram("Red fish, blue fish red fish")
{'red': 2, 'fish': 3, 'blue': 1}
```

14. (Classtime) Maximum Drop in Stock Prices

Write a function called max_drop(prices) that takes a list of integers representing daily stock prices. It should return the maximum drop in price between two days, where the later day has the lower price.

Function Signature:

```
def max_drop(prices):
    """
    Takes a list of integers and returns the maximum drop in prices.
    Return 0 if prices never drop.
    """
    pass
```

Example:

>>> max_drop([100, 80, 60, 70, 60, 75, 85])

15. (Classtime) Present Value of Multiple Cash Flows

You are a financial analyst at a mid-sized investment firm. One of your clients is considering an investment that will give them multiple future cash flows. They want to understand how much these future amounts are worth in today's terms using a fixed discount rate of 8%.

Theory: The Present Value (PV) of a future cash flow is calculated as:

$$PV = \frac{FV}{(1+r)^n}$$

where:

- FV: Future Value (amount of money to be received)
- r: Annual Discount Rate (in decimal)
- n: Number of years until the cash flow is received

When there are multiple future cash flows, the total present value is the sum of each individual PV.

Task:

Write a Python function total_present_value(cashflows, rate) that takes:

- cashflows: a list of tuples, where each tuple is of the form (amount, year)
- rate: the annual discount rate (as a decimal)

Your function should return the total present value of all future payments, rounded to two decimal places.

Function Signature:

```
def total_present_value(cashflows, rate):
    """
    cashflows: list of (amount, year) tuples
    rate: annual discount rate (e.g. 0.08 for 8%)
    Returns total present value rounded to 2 decimal places
    """
    pass
```

Example:

```
cashflows = [(10000, 2), (15000, 5)]
rate = 0.08
print(total_present_value(cashflows, rate))
# Expected output: approximately 20377.69
```

Note: You may assume that all inputs are valid, and the discount rate is non-negative.

16. (Classtime) The alternating harmonic series is given by:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \cdots$$

This series converges slowly to a value approximately equal to ln 2. You are asked to compute 'rearrangements of this series' by changing the order in which positive and negative terms are added. The rearrangements are explained below with examples.

Write a function rearranged_series(N, p, q) that takes three inputs:

- N: the total number of terms to use in the sum (starting from $\frac{1}{1}$)
- p: how many positive terms to include consecutively
- q: how many negative terms to include consecutively

Your function should:

- Always use the terms in the natural order $\frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \dots$
- Alternate between adding p terms and subtracting q terms
- Stop once exactly N terms have been used (not necessarily after a complete group)

Function Signature:

```
def rearranged_series(N, p, q):
    """
    Returns the sum of the first N terms in the harmonic series,
    rearranged to add p positive terms followed by q negative terms.
    """
    pass
```

Example 1:

rearranged_series(6, 2, 1)

This means: Add 2 positive terms, subtract 1 negative term, repeat. Use 6 terms total. Terms used in order:

$$+1 + \frac{1}{3} - \frac{1}{2} + \frac{1}{5} + \frac{1}{7} - \frac{1}{4}$$

That is: Positives: $\frac{1}{1}, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}$ Negatives: $\frac{1}{2}, \frac{1}{4}$ Sum:

$$1 + \frac{1}{3} - \frac{1}{2} + \frac{1}{5} + \frac{1}{7} - \frac{1}{4} \approx 2.45$$

Example 2:

rearranged_series(6, 1, 1)

This means: Alternate one positive and one negative term, just like the original alternating harmonic series. Use 6 terms total.

Terms used:

$$+1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6}$$

Sum:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} \approx 0.616$$

Your function will be tested with various values of N, p, and q. Make sure it handles:

- Partial final groups (e.g., when N = 5 and p = 2, q = 2)
- Cases where p > q or q > p
- The first N terms only

Try rearranged_series(1000,1,1) and rearranged_series(1000,2,1). They are not supposed to be equal!

17. (Classtime) With the advent of AI bots that can generate fully functioning code on command, even non-coders can now design software to solve real-world problems — a phenomenon often referred to as vibe coding.

As a budding economics or finance undergraduate, outline a real-world problem that you personally care about or find interesting. Then, explain how you could use vibe coding to solve it.

Guidelines:

- Clearly describe the problem.
- Explain why the problem matters to you or to society.
- Outline a possible software solution, even if you can't code it yourself.
- Be specific about what the software/app/automation should do.
- The more creative and unique your idea, the higher you will score. So dont share your idea with others and dream up unique ideas motivated by your life.

Hint: You could think of tools that help students, automate tedious financial research, improve transparency in data, track habits, etc.