CO1010

Questions

Answer all questions. Each question carries 10 marks.

1. Write a function $y = \text{my\_cumsum}(x)$ where the function computes

$$y_k = \sum_{i=1}^{k} x_i$$

where $x_i \in [0, 1)$ is a random array of floating point values. Use different values of k to demonstrate the accumulation of roundoff errors. Note that different values of k implies a variable length of x_i array.

2. Consider the simple algorithm

Algorithm 1

```
x := 0.0;
while x ⩽ 2.0
        print x
x := x + 0.1
```

What values of x will be printed? Implement the algorithm in a program and check that the correct values are printed. If this is not the case, explain what happened.

3. Identify values of x for which the formulas below may lead to large round-off errors, and suggest alternative formulas which do not have these problems

a) $\sqrt{\text{x+1}}$ - $\sqrt{\text{x}}$

b) $\ln(x^2) - \ln(x^2 + x)$

c) $\cos^2 x - \sin^2 x$

4. Write a function *my_ make_ size10(x)*, where x is an array and output is the first 10 elements of x if x has more than 10 elements, and output is the array x padded with enough zeros to make it length 10 if x has less than 10 elements.

```
def my_make_size10(x):
    # write your function code here
    return size10
```

Examples of output:

my_make_size10(range(1,2))

# Output: [1,2,0,0,0,0,0,0,0,0]

my_make_size10(range(1,15))

# Output: [1,2,3,4,5,6,7,8,9,10]

5. Rewrite the function my_make_size10(x) without using if-statements (i.e., using only logical and array operations)? (Hint: One can use functions such as numpy.where, numpy.insert, numpy.append, etc.)

6. Create a column vector using 50 integer values between [10,100] so that there is no repititon of any value. Reverse the vector.

Example Input: [10,25,11,24,14,16], Example output: [16,14,24,11,25,10]

7. Write a my_checkerboard() function to create a $n \times n$ checkerboard pattern consisting of alternate 1's and 0's without using the numpy.tile() function. Compare your code with the numpy.tile() function. Use $n > 100$ to compare your function with the builtin numpy function.You can use %%timeit magic to compare the time of execution.

Example:

$$\text{my\_checkerboard(4,4)} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

8. Normalize a 10×10 matrix containing random integer values between [-20,20] using the maximum value and the minimum value of the array such that all elements of the normalized matrix have values between 0 and 1.

9. Create two 2D (two-dimensional) numpy arrays. Write a function to find common values between two arrays.

10. Create a tuple. Write a program to find the index of an item within the tuple.