KOM1012 Spring 2025

Exercises

1. Write a MATLAB script to sum all prime numbers less than 1000. You can use the built-in *isprime* function.

2. Write a MATLAB function that duplicates each element of a vector. For example, for an input vector of [1 4 3], the function should return [1 1 4 4 3 3].

3. Write a MATLAB script that allows the user to input numbers as long as they enter positive numbers. Calculate the sum of all the numbers entered. Once a negative number is entered, stop the program and output the sum.

4. Write a MATLAB function that takes as input an integer N, and returns a vector containing the first N elements of the Fibonacci sequence.

5. Write a MATLAB script to find the first element of the Fibonacci sequence with 6 digits.

6. Write a MATLAB script to display all perfect numbers less than 10000. A perfect number is defined as a number for which the sum of its divisors is equal to that number.

7. Construct a finite sequence by performing the following arithmetic operations:
    - Pick a positive integer N. This is the first term of the sequence.
    - Then find the next term as follows: If the previous term is even, divide it by 2. If it is odd, multiply it by 3 and add 1.
    - Keep doing this until the term you get is equal to 1.

    The Collatz Conjecture states that no matter what number you start with, you will always end up with the number 1 in that sequence. The mathematical proof is not known, but no one has found a number that does not satisfy this property. A proposition like this is known as a Conjecture in mathematics.

    Using MATLAB, calculate how many steps you need to reach the number 1, for all starting terms from 2 to 1000. Which number gives the maximum number of steps?

(Adapted from Midterm Exam of Spring 2024)

A thief breaks into a store to steal some items. Fortunately for him, the value and weight of each item are written on a label on the item. They are shown in the table below:

Item no	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Value ($)	79	32	47	18	26	85	33	40	45	59
Weight (kg)	85	26	48	21	22	95	43	45	55	52

The thief has a bag that can hold only 120 kg of weight. Your goal in this problem is to write a MATLAB program to help him select the items that maximize the total value and can fit into the bag. This is an example of a well-known problem called the **0-1 knapsack problem**.

One way to solve this problem is to consider all possible combinations of items (for this example there are 2^{10} = 1024 of them), calculate the total weight and value for each possible selection, and find the one with the maximum value that does not exceed the capacity of the bag.

The function below solves the general problem rather than this numerical example. It inputs the weight vector, the value vector, and the capacity W, and outputs the best item numbers and their total value. **Complete this function for a general solution of this problem using the method explained. Then, find the best combination of items that fit in the bag for this numerical example.**

```
function [bestTotalValue, bestItems] = select_items(weights, values, W)

N = length(weights); % assume weights and values have the same length
items = 1:N;
I = genZeroOne(N);  % assume this exists, read the gray box below
bestTotalValue = 0;

for k=1:size(I,1)
    selected = I(k,:); % can be considered as logical or numerical vector









end
```

Information for 2024 Spring Midterm Problem

Consider a vector of length N whose elements are either 0 or 1. Let's call it a 01-vector. Consider all possible 01-vectors of length N. Of course, the total number of all possible 01-vectors of length N is 2^N.

In MATLAB, assume that there is a function called *genZeroOne* that takes N as input, and generates every possible 01-vector of length N as a row in a matrix of size 2^NxN. Two outputs are given on the right as an example. You can assume that the values in the matrix are logical or numerical type. We use this function in the problem above.

```
>> genZeroOne(2)

     0     0
     0     1
     1     0
     1     1
```

```
>> genZeroOne(3)

     0     0     0
     0     0     1
     0     1     0
     0     1     1
     1     0     0
     1     0     1
     1     1     0
     1     1     1
```