University of Science and Technology Houari Boumediene
Faculty of computer science

Algorithmics and Data Structures (S2) **INGENIEUR.INFO:** 2023/2024

TD N° 1: Subroutine (In class)

Exercise 1:

Write the Subroutine (procedure or function) to resolve the following problems:
1- Calculation of the factorial of N (N ! = 1x2x3x...xN, with 0 !=1).
2- Calculation of the sum S=1+2+... +N.
3- Calculation of the Maximum between two integers A and B.
4- Calculation of the n^{th} power (n >= 0) of a non-zero positive real number.
5- Calculation of the number of even digits in an integer N.
6- Calculation of the quotient and the remainder of the integer division of an integer A by an integer B.
7- **Optional:** Try to solve the previous problems using Recursion

Exercise 2:

1- Write a Subroutine "**Permute"** allowing you to swap two characters.
2- Let CH be a String. Using the previous Subroutine, write an algorithm to reverse the String CH.

Exercise 3:

1- Write two functions allowing you to calculate respectively the Greatest Common Divisor (GCD) and the Least Common Multiple (LCM) of two non-zero natural numbers.
2- Let T be an array of N non-zero natural integers, ($2 \leq N \leq 50$). Using the previous functions, write an algorithm allowing to:
- Display the GCD and LCM of the elements of T.
- Display all prime pairs of T.

Exercise 4:

1- Write a **Mirror** function to return the mirror of a natural number.
(example: Mirror(23568)=86532)
2- Write an **IntFrac** procedure allowing you to calculate the integer part and the fractional part of a real number.
(example: X=235.2601, integer part = 235, fractional part=0.2601)
3- Write a **Fexpo** procedure allowing you to transform a fractional part into the exponential form ($M \times 10^n$, with M >= 0).
(example: F=0.2601, M=2601 and n=4).
4- Let T be an array of N non-zero natural integers, ($N \leq 50$). Using the previous Subroutines, write an algorithm to display elements whose integer part is the mirror of the fractional part.
(example: X=23658.85632)

Exercise 5:

- Write a Subroutine **SYM** allowing you to check whether a square matrix of order N is symmetrical ($N \leq 20$).
- Let A be a matrix of NxN integers with $N \leq 20$. Write an algorithm that fills this matrix and checks if it is symmetric using the Subroutine SYM, and, in this case, displays the non-duplicate values as well as their respective positions.

1	3	7	5	2
3	-1	2	1	-2
7	2	2	6	0
5	1	6	8	-5
2	-2	0	-5	-2

Complementary Exercises (if we have time)

Exercise 6:

Write a parameterized ANAGRAM action that checks if two words are anagrams. Knowing that a word is said to be an anagram of another word if they are made up of the same letters.

Examples:

DOG is an anagram of CHINA, NICHE,

FREEZE is not an anagram of ALGER

Exercise 7:

1- Write a **DecToBin** Procedure which allows you to convert a positive integer into a string of binary characters ('0' or '1') representing its Binary code.

2- Write a **BinToDec** Procedure which allows you to convert a binary character string ('0' or '1') representing a Binary code in an integer.

3- Write an **XOR** Function which allows you to calculate the exclusive or (XOR) between two binary characters, we recall that:

A XOR B = '0' if A=B else it's '1'

4- Write a **BinToGray** Procedure which allows you to convert a string representing a code Binary into a string representing the equivalent Gray code.

5- Write a **GrayToBin** Procedure which allows you to convert a string representing a code of Gray into a string representing the equivalent Binary code.

6- Using the previous Subroutines, write a transcoding algorithm which, following a choice given as input (Decimal, Binary, Gray), displays the two other equivalent codes.

Exercise 8:

A number is called prodigious if it is divisible by the product of its non-zero digits.

Example: A=2016, 2x1x6=12 and 2016 are divisible by 12.

1- Write a **PRODIGIOUS** Subroutine that checks if an integer A is prodigious.

2- Let M be a square matrix NxN integers ($N \leq 50$). Write an algorithm that replaces the prodigious elements of the diagonal with the sum of the elements of the corresponding line, then displays the matrix if it has undergone modifications.