



Level: 1^e year computer science
Material: ADS2

TD/TP series No.: 01

Academic year: 2024/2025
Chapter 1: Subprograms

N.B. always write the main program that tests the subprograms, and the solutions must be in C.

Exercise 1: (TP)

- Create the "displayNbs" subroutine to showcase on the screen numbers that fall below a specified limit.

Exercise 2: (TD)

- Develop a "mirror" subroutine that accepts a natural number and presents its mirrored image on the screen. For instance, if the input is 5973, the program will show 3795.

Exercise 3: (TD/TP)

- Write the "displayTab" subroutine to showcase the elements of an array.
- Write the "sumTab" subroutine to calculate and return the sum of elements in an array.

Exercise 4: (TD)

- Write a "min" **procedure** that returns the minimum value between two real numbers using the following approaches:
 - Utilizing a **global** variable
 - Employing "variable passing"
 - Rewrite this procedure as a **function**

Exercise 5: (TD)

- Define a structure to hold the coordinates of a **point** (x, y).
- Write a subprogram "norm" to calculate the norm of a vector.

Exercise 6: (TP)

- Write the "isSeparator" subprogram to determine whether a character is a separator or not. The separators include (., !, and space).
- Write a "countWord" subroutine to calculate the number of words in a sentence.

Exercise 7: (TD)

- Write the "fact" subroutine to compute the factorial of a number.
- Write a "power1" subroutine to calculate -1 to the power of y.
- Write a "power" subroutine to compute x to the power of y.
- Write the "cos" subprogram to evaluate the following sum:

$$S = \sum_{i=0}^n \frac{(-1)^i x^i}{(2i)!}$$

Exercise 8: (TP)

- Write a function named "isPrime" to determine whether a given number is prime or not.
- Additionally, if you are aware that any even number is the sum of two prime numbers, create a program that takes a number 'n' as input and displays all pairs of prime numbers whose sum equals 'n'.

Exercise 9: (at home)

- Create the "isPerfect" subroutine to determine whether a number is perfect or not, considering a perfect number as one that equals the sum of all its divisors except 1 and itself.
- Write a program to display all perfect numbers less than N.

Exercise 10: (at home)

- Write an "inverseTab" subroutine to exhibit the elements of an array in the reverse.
- Write the "strlen" subprogram to calculate the length of a string s.
- Write the "pos" subroutine, which returns the location of the symbol x in the string s starting from position d and returns -1 if it doesn't find it.