

*Objective of the session :*

Perform top-down analysis and detailed design of a program that manipulates arrays.

## 1 Top-down analysis of the Mastermind game

Perform the top-down analysis and then the detailed design to play the Mastermind game opposing 2 players. The rules of Mastermind are as follows :

- The first player chooses an orderly combination of 4 colours from among the authorized colours : yellow, orange, pink, red, green, blue. This combination is not known by the second player who will have to guess it
- The second player has a maximum of 10 attempts. Each of her attempts is as follows :
  1. the second player proposes a combination of 4 colours
  2. the first player responds with a number of black pawns and a number of white pawns. The number of black pawns corresponds to the well-placed colors. The number of white pawns corresponds to the colors present in the combination but present at another place. **Warning** : a color at a given place cannot lead to the appearance of more than one pawn (whether black or white).
- The game ends when the second player finds the hidden combination (she wins) or when she reaches the end of her attempts (she loses).

### 1.1 Global design

Reminder : to perform a top-down analysis, follow the four steps below :

---

#### *Exercices*

---

1. Informally define the tasks to be performed by the program, and possibly sub-tasks within each task.
  2. Write this functional decomposition in the form of a top-down analysis diagram.
  3. Identify the inputs and outputs of each subprogram, defining if it is necessary your own types (here to represent a color, a combination, etc.)
  4. Choose to represent each subprogram with a function and write its signature.
-

## 1.2 Detail design

---

### *Exercices*

---

1. Write the detailed design of the functions
  2. Write the detailed design of a main program
- 

## 1.3 Additional handling (on machine)

---

### *Exercices*

---

1. Implement the previously defined program in pascal,
  2. Add a subprogram that replaces player 1,
  3. Split the types and functions definitions in separate and coherent units.
  4. Save the objective combination and the execution of the current game in a file.
  5. Display the last round at the beginning of the program.
-