# PW 4: Group Work

## Goal

- Group work
- Write and test subprograms (always!)
- Write explanations (always!)

## Instructions

This exercise will be carried out in teams of 4 students following this schedule:

1. 10 minutes: individual reading of the whole subject,
2. 20 minutes: carrying out exercise 1. After an individual reflection of 5 minutes, the individual ideas are pooled and discussed to arrive at a common solution which will be adopted by the whole group for the rest of the work.
3. 30 minutes: specification of the subprograms (exercises 2 to 9)
    a. 5 minutes: the team members distribute the exercises from 2 to 9 (2 to 3 exercises each). Exercises 3, 6 and 9 must be handled by different team members.
    b. 10 minutes: each member of the team writes down the specification of the subprogram assigned to them and identifies the tests that should be done.
    c. 15 minutes: presentation of the specifications to the other team members and discussions to improve them.
4. 45 minutes: refining the subprograms
    a. 15 minutes: Each team member writes the first refinement (algorithmic thinking) levels of the sub-programs which have been assigned to him.
    b. 30 minutes: Presentation of refinements to other team members and discussions to improve them.
5. 45 minutes: Each team member writes on a machine the specification and implementation of the subprograms assigned to them. He also writes their test programs.
6. 30 minutes: exchange of individually written subprograms to write the complete program.

## Connect 4

The game "Connect 4" is offered by MB Games. On the box, we can read the following indications.

Connect Four is easy to learn and fun to play. Its rules are simple:

- It is a two-player connection board game,
- in which the players choose a color
- and then take turns dropping colored discs into a seven-column, six-row vertically suspended grid.
- The objective of the game is to be the first to form a horizontal, vertical, or diagonal line of four of one's own discs
- while trying to prevent their opponent from doing the same.

Rules of the game :

- Choose the first player. The player who starts the first game will be the second player in the second game.
- Each player takes turns dropping one of their discs into one of the slots at the top of the grid.
- Play continues until a player has a continuous alignment of four discs of his color. Alignment can be vertical, horizontal or diagonal.
- To clear the grid, push the retaining bar at the bottom of it and the discs fall. You can now start the next part.

## Exercise 1: Preparing for the game

The game "Connect 4" is played on a vertical grid of six rows and seven columns with 21 red discs and 21 yellow discs. Define the types needed to represent the game grid.

## Exercise 2: Clear the grid

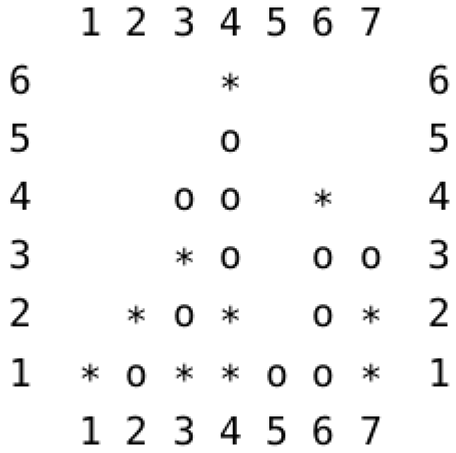
To start each game, you must first empty the grid of discs it contains. Write a subprogram that does this.

## Exercise 3: Show the grid

Write a subprogram that displays the contents of the grid;

- red discs will be represented by (*)
- yellow discs by the lowercase letter o.

The grid will be like:



```
  1 2 3 4 5 6 7
6       *       6
5       o       5
4     o o   *   4
3     * o   o o 3
2   * o *   o * 2
1 * o * * o o * 1
  1 2 3 4 5 6 7
```

## Exercise 4: Deciding if a move is possible

Write a subprogram that indicates whether it is possible to play in a given column.

Example: In the previous figure, it is possible to play columns 1, 2, 3, 5, 6 or 7. It is not possible to play column 4.

## Exercise 5: Drop a disc

Write a subprogram that does this operation: it calculates the new state of the grid when a disc is dropped above a given column.

Example: Starting from the previous figure, if red plays column 5, the new state of the grid is:

```
    1 2 3 4 5 6 7
6         *        6
5         o        5
4       o o   *    4
3       * o   o o  3
2     * o * * o *  2
1   * o * * o o *  1
    1 2 3 4 5 6 7
```

## Exercise 6: Count the aligned disc

Write a subprogram that indicates what is the length of the longest alignment of a given disc (identified by its column number and row number). Alignments should be checked in all directions (horizontal, vertical or diagonal).

This subprogram will then be used to determine the end of a game (alignment ≥ 4) or to help the computer choose which column to play (exercise 8).

In the example of the previous figure, box (1,1) corresponds to an alignment of three discs diagonally; box (7,1) corresponds to an alignment of two discs vertically; the box (6,2) corresponds to three discs aligned vertically or diagonally …

## Exercise 7: Recommend a column to play in

Write a subprogram which, given a grid, offers a column in which to play. The column number will be chosen at random.

## Exercise 8: Improve advice

To improve the advice, we add an additional constraint: advise the column that allows the longest alignment.

Note: If a winning move exists, the advised column will lead to victory.

Example: Starting from the previous figure, the yellows will play in column 5. This causes an alignment of 4 discs and therefore victory.

## Exercise 9: Play a game

Write a program that let a human player play against the computer. The human player is considered to have the yellow chips and begins the game.