

General rules

Students are requested to perform and deliver exercises on a **weekly** basis.

The delivery of the exercises will account for a maximum of 0.5 points over the final mark. In order to be marked, at least 80% of the weekly exercises must be delivered.

The next lab class after the delivery of weekly exercises, random students will be asked to **orally** present the solution to the delivered exercises. A maximum of 0.5 points will be assigned to this item.

Delivery will be individual.

Even if the marks assigned to the practical exercises can seem small and despite they are not compulsory, performing them is essential to gain the knowledge needed to pass the subject. It is highly recommended to perform and deliver the weekly exercises on time. The exercises of a week will be the basis for the new concepts that will be explained the following week.

Introduction to algorithms

An algorithm is a sequence of steps to solve a given problem. The algorithm to replace a flat tire shown at class is a good example. Next, an example of an algorithm to buy a metro ticket on a ticket machine is shown:

1. Go to the machine
2. Choose which kind of ticket you want
3. Pay
4. Go to the door

This algorithm belongs to the most simple kind of algorithms; sequential ones. In sequential algorithms, steps are executed in sequence, one after the previous one. But, except for quite simple tasks, sequential algorithms are not really useful and conditions and iterations must be added.

Next, an example of an algorithm including conditions is shown. It represents a more complex procedure for buying a metro ticket:

1. Go to the machine
2. Choose which kind of ticket you want
3. Do you want to pay with cash?
 - 3.1. Look for coins
 - 3.2. Pay with cash
4. Else
 - 4.1. Pay with credit card
5. Did you pay?
 - 5.1. Go to the door

In this case, step 3 determines if either the steps 3.1 and 3.2 or the step 4.1 will be executed. We say that step 3 is a condition with an “else” part: if the condition is true something will be executed, if it is not true

another thing will be executed. Something similar applies for step 5, if the condition is true something will be executed, but in this case nothing is executed if the condition is false.

In addition to conditionals complex algorithms usually include loops. An example of an algorithm including loops is the following:

1. Go to the machine
2. Choose which kind of ticket you want
3. Repeat until ticket has been paid
 - 3.1. Do you want to pay with cash?
 - 3.1.1. Look for coins
 - 3.1.2. Pay with cash
 - 3.2. Else
 - 3.2.1. Pay with credit card
4. Go to the door

In this case, step 3 is a loop; it will be repeated until the ticket is paid.

Exercises

- Exercise 1.** Describe the algorithm to cook a potato omelet.
- Exercise 2.** Describe the previous algorithm, but take into account that some people do not add onion to the potato omelet.
- Exercise 3.** Describe the previous algorithm taking into account that 5 eggs need to be cracked and that they have to be cracked in sequence.
- Exercise 4.** Describe the algorithm you follow to get money from an ATM.
- Exercise 5.** Extend the previous algorithm so the ATM checks if the user has money enough in the account. If no enough money is available, the ATM will do nothing.
- Exercise 6.** Extend the previous algorithm so if there is no enough money, the ATM asks the user to introduce a new amount (it will keep doing it until the user enters a correct amount)
- Exercise 7.** Create an algorithm that given any two numbers A and B, calculates if A is divisible by B, using just additions, conditionals (comparisons) and loops.

Delivery rules

The solutions to the previous exercises must be uploaded to Aula Global. Each student is requested to upload a zip file containing a PDF file for each exercise (7 PDF files compressed in one zip folder). The name of the zip file must be "w1-name-initials.zip" (for example if the student is Lucía Pérez Gómez the file will be named w1-lpg.zip)