# IATSED Python Programming - Exam

## Exercise 1:

Write a Python program that asks user to enter **positive** numbers until he/she enters a zero value. After that, the program should print the product and the numbers entered (before the zero). Handle situations when user enters non-numeric characters.

Example :

Enter a positive number (0 to stop) : 3
Enter a positive number (0 to stop) : -1
Please enter positive number!
Enter a positive number (0 to stop) : 5
Enter a positive number (0 to stop) : Andrija
Please enter positive number!
Enter a positive number (0 to stop) : 2
Enter a positive number (0 to stop) : 0
Results: 3 * 5 * 2 = 30

## Exercise 2:

Define a function **count_chars(text)** that takes one string argument and that returns a dictionary linking each alphabetic character (**case insensitive**) to its number of occurrences in the string.

## Exercise 3: Flight sequencing

For a given list of flights with their planned time of arrival it is required to build the arrival sequence by delaying flights such that separation norms due to wake vortex are satisfied, respecting at all time **First Come First Served** principle.

File **arrival.csv** provides a list of arriving flights (not necessarily sorted), containing their arrival time [**in seconds** from the beginning of th day] and aircraft type (and some other data).

Aircraft type’s wake vortex categories are given in **ac_categories.csv** file.

Wake vortex separation norms for different (preceding and succeeding) aircraft categories are given in table [**in minutes**].

| Preced. \ Succ. | Heavy | Meadium | Light |
|---|---|---|---|
| Heavy | 2 | 4 | 6 |
| Medium | 2 | 2 | 4 |
| Light | 2 | 2 | 2 |

Write a Python program that calculates and prints acceptable arrival sequence taking into account separation norms and aircraft wake vortex categories.

Flights should be printed using identifier, planned and calculated arrival time.