INSTRUCTIONS

1. **Keep calm and do the test**.

    You can finish the test and send it by the limit date explained by email. *This test was made to evaluate your logical reasoning and you should solve it in the simplest way.*

2. Once you have finished the test, you must send the answer to the emails **tatielle.rodrigues@artefact.com and caio.garvil@artefact.com.** Name the subject of the email like the example below:

    Example: Artefact - Written case - *Data Engineer*;

3. The essay answers can be sent in "**.pdf". ** The solution of the required algorithm **MUST** be sent in **".r", ".ipynb" or ".py".**

4. All the responses **must be in English!**

5. Finally, do not forget to name your document with the answers following the example below:

Your Name - Position - Seniority Level
Example: Caio Garvil - *Data Engineer- Junior*

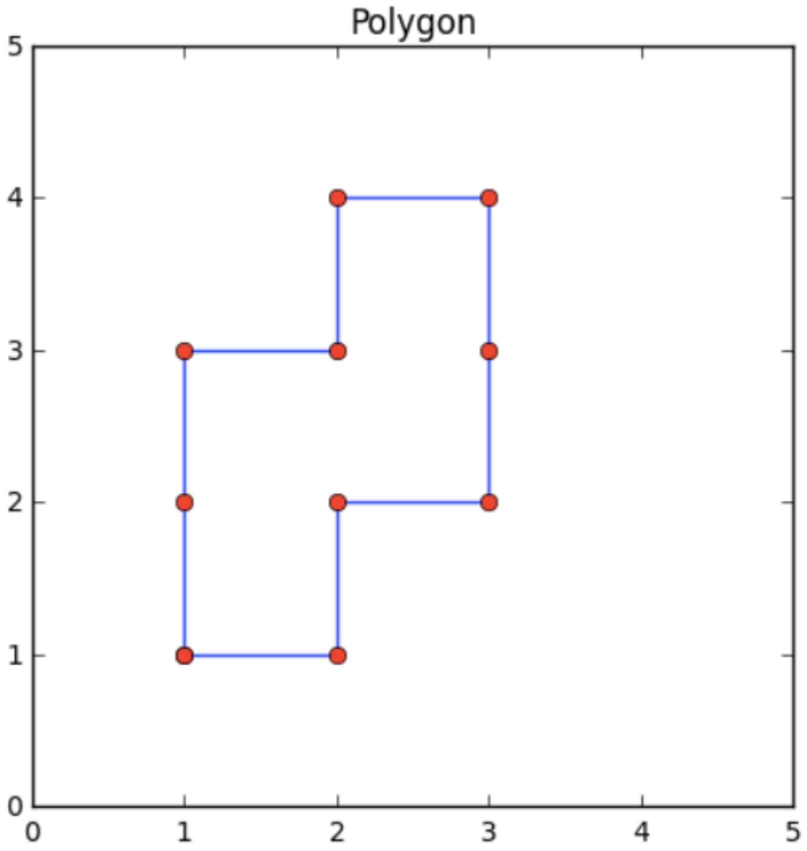
Data Engineering

Problem 1: Polygon Optimisation

Let us consider the following set of points :

points = [(1, 1), (1, 2), (1, 3), (2, 3), (2, 4), (3, 4), (3, 3), (3,2), (2,2), (2,1), (1,1)]

This set of points defines the following Polygon:



The goal of the exercise is to reduce the previous set of points to keep the necessary points to draw the same Polygon.

a) Find the points that are not needed to draw the Polygon and explain why.

The points not needed are (3,3) and (1,2) because those points are on the lines created by another 2 points.

b) Write a function that, given a Polygon's path of points, returns a reduced list that defines the same Polygon. (Notice that all the points in the set are adjacent). **Choose the language you want to answer this question, and comment on the code explaining each step. Don't worry about syntaxes.**

Problem 2: Incorrect syntax with brackets

a) Given a string made of opening and closing brackets, how to check if its syntax is OK? **Choose the language you want to answer this question and comment on the code explaining each step.**

Examples: ()(()()) is correct, ())) is not,)(is not

b) Follow-up questions **(code not needed, just explain what you would do)**

i) How would you do it with multiple bracket types (), {}, [], '', “”

For multiple bracket types, I would have multiple counters. Also, I would also check whether there were any brackets that closed with another open bracket inside.

ii) How to use that to correct real code syntax with not only brackets but letters, numbers etc? (e.g.: ((a) → (a))

If we’re only trying to check, then the current code works. However, to correct the strings, we could add any amount of unmatched open brackets “(“ with closed brackets “)” at the end and delete any excess closed brackets “)”.

iii) How to use what has been done to output the correct syntax?

This was answered in the previous question ii).

iv) How to develop a version of the algorithm that would output all possible valid strings?

Assuming that I’m provided with a list of strings. First, create a list of booleans using my syntax_checker function to apply over the entire list of strings. Then, filter the list of strings using the new list of booleans.

Problem 3: Software Architecture

Based on the concept of Microservices, APIs (REST, gRPC, etc…) and Databases, briefly describe (you can use words, flows, bullet points, diagrams, etc… what is important is that your explanation is as clear as possible, and specify the technologies that you will use) how would you apply this concept to create the following application requested by a client :

“I would like to create a visual tool that is able to Read, Write, Update and Delete a “To Do’s” checklist to organise my schedule. I would like to be able to integrate this with my Google Agenda as well, can you do that?”.

Visual:
Create an application using Flask. Write a HTML file and include Bootstrap to use the wide range of visual tools from Bootstrap.

In the App, we need a space to enter the task with due date. Also, a space to see the entries that are already there and edit (alter or delete) those tasks, if needed.

Integration with Google Calendar
We can integrate the application with Google Calendar using the Google Client library in python. This library has many methods to create, read, and delete event. To alter, the event would first need to be deleted then recreated with the alterations.