Problem Statement

As part of a large piece of work, Advertising Technology needs to develop a Filtering Proxy service to support Advertising Decisioning on the brand new Sky Glass platform.

This task is to build a **RESTful** service which determines if a targeted advert should be requested, or that the linear advert should be shown without a substitution.

The service endpoint receives the following in the request parameters :

customerId - numeric - 12 digit ID for the Sky Customer the TV is registered to
optIn - True/False - indicating if the Customer has opted in to targeted advertising
inactivityTimer - numeric - number of seconds since the Customer last used the remote

API will return relevant response codes (200/4xx/5XX) with response {'value' : True/False} based on the following logic:

If optIn is False, they have not given Sky permission to send targeted adverts so you must return False
If inactivityTimer is greater than 2 hours then they have left the room, or fallen asleep, and so we do not send them any more targeted ads, so return False
If customerId is exactly divisible by 7 then they are part of the market segment the linear advert is aimed at and so we do not want them to get a targeted ad as the original ad is already targeting them, so return False.
If inactivityTimer is exactly divisible by 9 then the downstream ad decision system is offline, so return False
Otherwise return True

You may use any of these languages - Python, Go, Java, C# - and any IDE/Tools you feel comfortable with.

As you can imagine, with something this central to AdTech's operations we need to ensure quality. We will be interested in how you can assure top quality.

For information, the system will have a full CI/CD pipeline (we do not expect you to develop a pipeline for it - but the quality checks would gate that pipeline)

During the interview we would want to discuss your approach and also as is true of all development work, describe to you a change in requirements which you would need to cater for.

What are we looking for during the test

- Demonstrating your Python/ preferrable language knowledge
- Demonstrating knowledge of and applying OOP skills

- Bonus point for creating an OpenAPI/Swagger document
- Use of industry standards for Restful APIs
- The use of frameworks to achieve the task
- Use of data structures, data modelling, code architecture
- Use of TDD, SOLID, DRY principles
- Self-documenting code
- Extensibility
- Error handling
- Efficiency
- Validation

Please do not spend more than 3 hours on this exercise - we are not looking for a perfect answer. We are looking more about how you approach the problem and the subsequent change in requirements.