

CoGrammar.com

— P O W E R I N G —

Hyperiondev.com

**Assessment
Python Developer**

www.CoGrammar.com

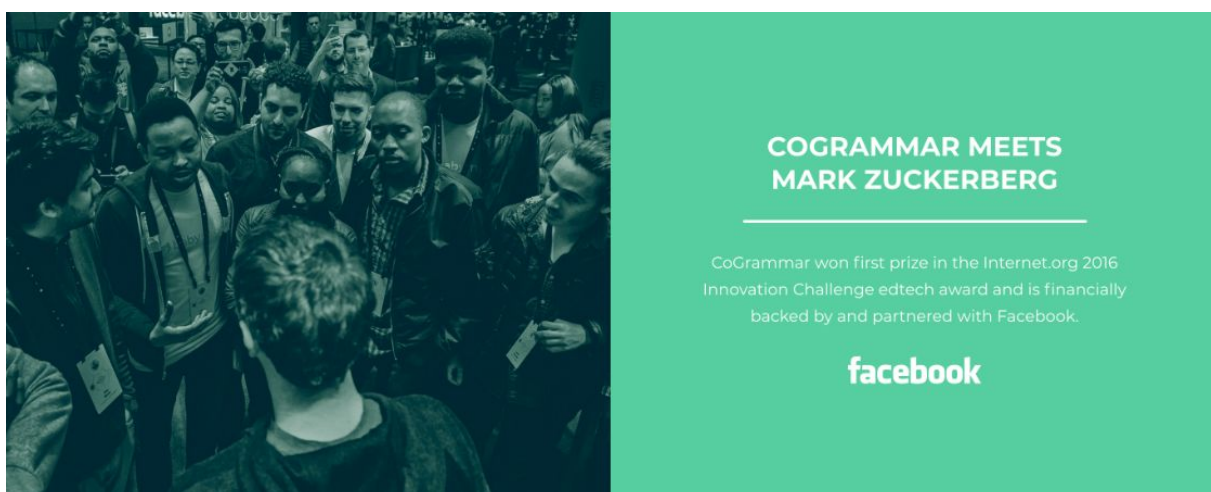
Welcome

Congratulations for being shortlisted for a role with CoGrammar! We're excited to get to know you and your skills better. The first step of the interview process with CoGrammar is to complete a take home exercise. Please complete this exercise within **10 calendar days** of receiving it. Please take note of the submission guidelines below in the How To Submit section below.

What is CoGrammar?

Cogrammar is recognised as one of the top education technology startups in Europe, Middle East, and Africa. Founded by [Riaz Moola](#) in 2012, CoGrammar's leadership team consists of ex Google, Amazon and PWC employees, senior team members from GetSmarter (most valuable edtech startup in South Africa, recently acquired for >R1.5 billion) and graduates of the University of Cambridge, Cape Town, and Oxford.

CoGrammar was funded by Facebook and Google in 2017 - winning first prize in Facebook's Africa Innovation Challenge Award as the top edtech startup on the African continent. CoGrammar works directly with Facebook's senior leadership - right up to Mark Zuckerberg himself - and through its headquarters in London is recognised as a leading edtech startup in Europe, the Middle East, And Africa. CoGrammar was recently recognised as one of the top 5 edtech startups in South Africa and is supported by top globally edtech investors including the edtech fund behind [Coursera](#), [Andela](#), [Udemy](#), & [SoloLearn](#) and



We have built an online course platform that allows human code review to be scaled, applying this methodology to help thousands of students from over 30 countries learn how to code in a novel way. We pioneer effective and affordable software development

education with this code review model, lowering the cost of access to tech careers around the world to shrink the tech skills gap and inequality in the tech space.

Ensure you have reviewed the [about us](#) page before continuing.

Being a python developer at CoGrammar

CoGrammar is a tech-first company. Our founder and CEO has a degree in Computer Science and wrote some of the initial code himself. As a back-end developer in the Platform team, you will be an integral part of building a new platform for online education. You will craft high quality code to build an API from the ground up, integrating with third-party APIs along the way.

We work in a friendly environment, meeting weekly for structured sprint discussion and have ad-hoc discussions as issues arise. You will have weekly 1:1 time with the technical lead to help you work through any problems and to ensure that you continue to grow technically and professionally.

You will work with Django, Python, GitLab and AWS. In addition to coding, you will be involved in architecture, choice of technology, requirements specing, and you will gain a deep understanding of the entire technology stack and business.

You will be part of CoGrammar's mission to close the tech skills gap by training thousands of new software developers across the world. Your work will directly help aspiring coders, companies seeking talent, and the global economy.

What's more is we need to bring the brand to life, let the world know who CoGrammar is and where we fit in the world. It's more than selling online coding courses. We believe that people can change their lives and with the skills to become a developer go on and change the world.

Your tasks

This assessment consists of three tasks. Please submit all three of these tasks.

1. Write a Python script that builds Magic Squares.

Magic squares are described in full in Appendix A.

Your script should:

- Allow the user to enter N through standard input, where N is any positive odd integer.
- Check that N is valid and prompt the user to enter another value for N if it is not.

- Calculate a valid magic square for that value of N
- Print out the magic square, legibly
- Verify that the magic square is indeed valid
- Print “correct” if the verification step passes.

Please consider code readability to be the highest priority (apart from producing the correct solution).

2. Build a demo application that integrates with the Dropbox API

Create a Python project that uses the Dropbox API. You should use the V2 API <https://www.dropbox.com/developers/documentation/python> and the Python SDK provided by Dropbox.

Include a README file to explain what your app does and how.

You can do as little or as much as you want for this project. It's your chance to impress us, but we also understand that your time is limited. We will assess based on several factors including:

- Project structure and code
- Functionality
- The README file

3. Tell us what you consider important in software development

Write three or four paragraphs about how you would develop a good web API. Talk about the process you would follow from idea to release, the tools you would use, and the problems you would need to solve. You could talk about scalability, authorization, design, discoverability, and/or anything you think is important.

Task Submission

Please compose your response to these tasks in a new document and email the document to the email ID from which you received this assessment, along with your CV. Thank you for your interest in joining our team at CoGrammarDev!

If you need to send large files as part of your submission, please upload these to Google Drive or Dropbox and include a link in your email. If you want to submit your code through GitHub or similar, please make sure you use a private repository.

Appendix A:

Magic Squares

A magic square is a square grid of distinct numbers such that each row and column add up to the same number. Further, the two diagonals (from corner to corner) also add up to that number.

An example is the 3x3 magic square (from [Wikipedia](#)) which is shown below:

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↓15
			↘15

Note how the arrows indicate the sum of each row, column, and diagonal. For the purposes of this task we will only be looking at creating a magic square ($n \times n$ matrix) where n is odd, e.g. 3x3, 5x5, etc. This is due to the approaches being slightly different depending on the value of n .

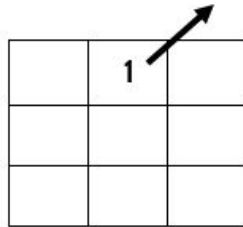
The logic for an $n \times n$ magic square, where n is odd

The logic may seem slightly complex for now, but you may find it easier to understand if you follow the logic whilst looking at the solution given above (the 3x3 matrix). Each step you take you fill in the next number, e.g. if you last entered 3, you should now enter 4. The steps are as follows:

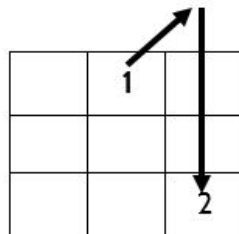
- Start with a 1 in the middle column of the first row.

	1	

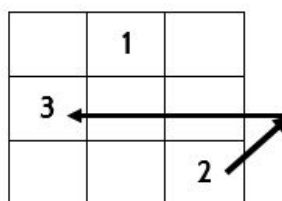
- Attempt to move diagonally up and right by one block.



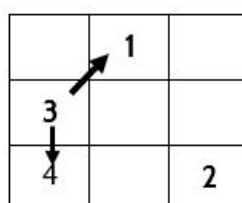
- In this case you can't - you have gone off of the top of the grid. When you go off the grid when attempting to go diagonally up and to the right by 1 block, then wrap around to the other side of the grid where it would go in the equivalent row or column. Think of the old cellphone game of snake when you went through the edges!



- You then go back to trying to go diagonally up and to the right, which again you can't, so this time you wrap around to the left.



- Again, you try and move diagonally up and to the right by 1 block. This time you can't, but not because of going off the grid, but because you bump into an existing number. In this case, you simply go down by 1 block.



- From there, we try to move diagonally, which we can twice!

	1	6
3	5	
4		2

- This leads us to a more interesting case of going off the edge of the grid - now we go off diagonally! So to solve this we go down by 1 block.

	1	6
3	5	7
4		2

- Once again, we try and go diagonally up and right by 1 block, which we can't. So we wrap around to the left.

8	1	6
3	5	7
4		2

- Although it's fairly obvious where the next number should go, let's continue with our method. Try and go diagonally up and right by 1 block. We can't - we're off the grid, so we wrap down to the bottom.

8	1	6
3	5	7
4	9	2

And we're done! Confirm that each row adds up to 15! This method works for all odd $n \times n$ magic squares! Now it's your turn to generate this solution in code.