

TAKE HOME TEST - WRITTEN ASSESSMENT

# Junior Software Engineer

2020

# Welcome

Congratulations for being shortlisted for a role with [HyperionDev](#)! We're excited to get to know you and your skills better.

The first step of the interview process with HyperionDev is to complete a take-home exercise. Please complete this exercise within **5 calendar days** of receiving it and make sure your responses are all sent through to the email ID from which you received this assessment, unless otherwise specified.

## What is HyperionDev?

HyperionDev is the largest provider of software development education in Southern Africa, and one of the largest globally. Founded by [Riaz Moola](#) in 2012, HyperionDev is counted among the top education technology startups in EMEA with headquarters in Cape Town and London. We were originally launched from the University of Cambridge, and our leadership team consists of world-leading talent from companies such as Google, Amazon, 2U/GetSmarter, Yoco and PwC.

HyperionDev 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. We were recently recognised as one of the top 5 edtech startups in South Africa and are supported by top global edtech investors including the edtech fund behind [Coursera](#), [Andela](#), [Udemy](#), & [SoloLearn](#).



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 40 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. Please ensure you have reviewed our [about us](#) page as well.



Page 3



## Your 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.

Page 4



## Task Submission

Please compose your responses to the above tasks in a new document and mail the document back to the email ID from which you received this assessment. Alternatively, you may use Google Docs and send us the links in your email.

You would also be able to respond in email with all questions you have around the assessment. The hiring manager will be able to then provide you some guidance, resources and answers (if possible) around your query. The aim is for you to succeed!

This assessment is designed to understand you better as a candidate and has no right or wrong answers. We will be weighting the assessment based on the following:

1. Thoroughness of your submission
2. Use of the data provided
3. Creativity

- 4. Research efforts
- 5. Presentation
- 6. Completed within given time

**Thank you for your interest in joining our team at  
HyperionDev!**

Page 5

Squares



## **Appendix A:** Magic

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

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.  $3 \times 3$ ,  $5 \times 5$ , 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

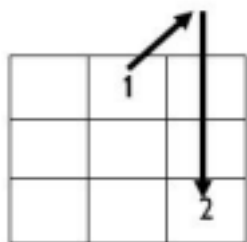
The logic may seem slightly complex for now, but you may find it easier to understand if you follow the logic whilst looking at the solution given above (the  $3 \times 3$  matrix). Each step you take you fill in the next number, e.g. if you last entered 3, you should now enter 4. The steps are as follows:

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

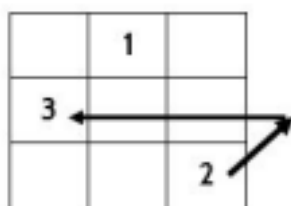
	1	

- 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

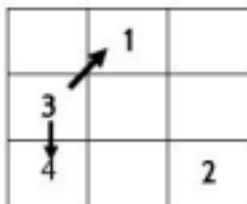
it would go in the equivalent row or column. Think of the old cellphone game of snake when you went through the edges!



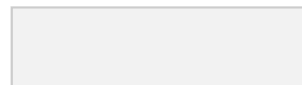
- You then go back to trying to go diagonally up and to the right, which again you can't, so this time you wrap around to the left.



- 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.



Page 7

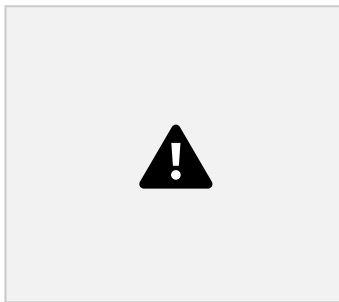


- From there, we try to move diagonally, which we can twice!

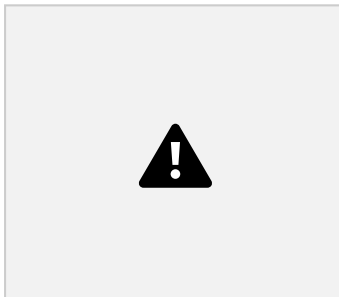




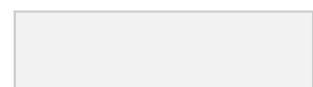
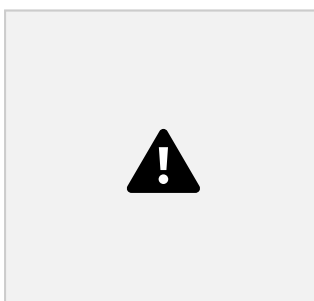
- This leads us to a more interesting case of going off the edge of the grid - now we go off diagonally! So to solve this we go down by 1 block.



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



- 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.



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.