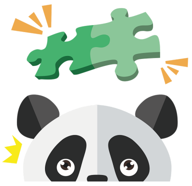
Developer Development task - Puzzle



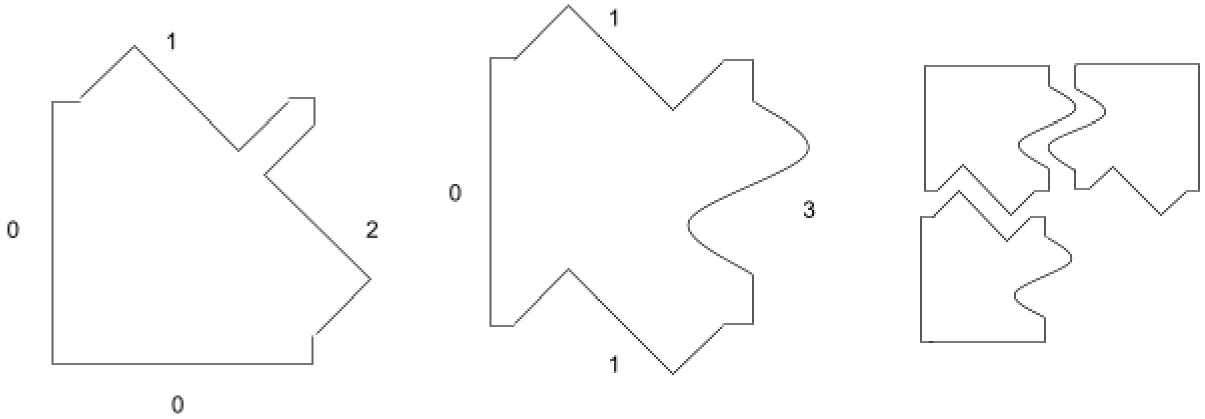
This task is a complex problem to assess your abilities as a developer. It is designed to allow you to demonstrate some skills that we are looking for in a successful candidate, particularly, your ability to write efficient, clean, maintainable, generic algorithms.

Exercise:

The task is to create a script that will solve a puzzle.

All the pieces of our puzzle have four sides. Each one of the sides is represented by a positive integer. These numbers represent shapes. The number zero represents the border, which is a special side.

For example, on these pieces, you can see sides 0 (border), 1, 2 and 3. Side 1 fits with side 1, side 2 would fit with side 2, etc. On the last images you can see how 3 pieces could fit together.



Even though the sides 1 and 2 look similar, they are actually mirrored and that's why they have a different number.

There could be more than 2 pieces with the same side. That means that those pieces can fit together.

The sides of the pieces are named sequentially, clockwise, starting with the left side. For example, the first piece is 0 1 2 0.

The puzzle file is defined on a text file. A puzzle has a width and a height, which are defined on the first row of the file; then, line by line represents each one of the pieces. For example, this is a 16-piece puzzle:

```
4 4
2 5 4 0
2 1 4 2
0 1 1 0
4 4 0 3
0 0 4 3
0 0 1 1
1 4 0 0
4 4 3 5
5 5 2 4
1 1 0 5
4 1 0 4
1 0 2 4
3 5 1 2
1 4 2 0
0 1 5 2
1 5 0 4
```

Each piece is numbered sequentially, and the first piece is the piece number 1.

This is a solution to that puzzle:

```
3 10 16 7
15 13 2 11
12 8 9 4
6 14 1 5
```

The solutions are written using the piece numbers, separated by spaces, and one line per row.

A puzzle could have multiple solutions. For example, this is also a solution for that same puzzle:

```
3 12 15 6
14 8 13 10
1 9 2 16
5 4 11 7
```

Actually, all puzzles will have more than one solution, since you can rotate the solution 90/180/270 degrees. For the purpose of this exercise we are not interested on the rotated solutions. With that in mind, the puzzle above has exactly two solutions. Different solutions are represented as below, with a new line in between. These would be the solutions file for the sample puzzle:

```
3 10 16 7
15 13 2 11
12 8 9 4
6 14 1 5

3 12 15 6
14 8 13 10
1 9 2 16
5 4 11 7
```

Rules:

- You need to use all the pieces and you can only use each piece once, (please note that there could be two identical pieces listed)
- The border needs to be around the puzzle
- The corner pieces have to be at the corners
- You **can rotate** the pieces. For example, if a piece is 0 1 2 3, you can rotate it, i.e. that piece is the same as 1 2 3 0, 2 3 0 1 and 3 0 1 2.
- You **cannot flip** the pieces. For example, the piece 0 1 2 3 from the previous example is not the same as 0 3 2 1, so you cannot use it as that.
- The solution ideally needs to be written in PHP*

* Please note that for some of our vacancies, solving this task in PHP is not a requirement. If you are not sure if you have to do it on PHP, and you prefer to solve it in a different language, please ask us.

Your solution needs to take a file as an input (e.g. argument), and output the solution after processing it. For example, a call from the command line, could be `php solve.php 4x4.txt` and will output the solutions above. Your solution doesn't need to be contained in a single file, it could include multiple classes / files. If you wish, you can use additional tools as long as you call them from the main file.

Your solution will be tested with other puzzles, so you need to make sure that your algorithm works on every scenario. As a sample, we are providing a 5x5 and a 8x8 puzzles for you to test your script. The 5x5 puzzle has one solution, the 8x8 puzzle has 2. Your program should provide all the solutions, not just the first one.

Please note, your solution doesn't need to have a front-end, unless you want to. A command-line script is fine.

Note about Javascript: If you have chosen to use Javascript to solve this problem, reading the files from disk is not straightforward. In this case our advice is that you copy-paste the contents of the input files into your code, so you can focus on the main problem and not how to read a file from disk.

Marking criteria:

The exercise will be scored out of 10 marks. This is the breakdown of the marks:

- Clean, readable, elegant code (1 mark)
- Performance (memory usage and resolution time) (2.5 marks)
- Resource usage (2.5 marks)
- Any other significant enhancement not listed on the requirements (2 marks)
- Explanation of the approach (1 mark)
- Lessons learnt, WWW, EBI and how this algorithm could be improved in the future (1 mark)

Please note that the first four points have to do with code, and the last two don't. They are about written responses.

Submission:

Please **zip the files** and reply to the email we sent you to send them to us. Just give us an example of how to call the command line. Please, **do not** submit your code to a public repository, as there are other candidates in this selection process that could find it. Please, **do not** post this exercise online anywhere. Other candidates on the selection process might find it and have an unfair advantage.

If you used a "compiled" language, please remember to include the source code files, for example the .java files and not just the .jar; or the .cpp and not just the binary. If you are sending a .js file, Google will block it, so please rename it to .txt before zipping it.

Additionally, please provide answers to the following questions:

- Explanation of the approach taken (1 mark)
- Lessons learnt (0.25 marks)
- What Went Well (0.25 marks)
- Even Better If (0.25 marks)
- How could you improve the algorithm in the future (0.25 marks)
- How long did you spend solving this problem? (No marks)
- Are there any changes that you think we should make to these instructions or anything that we should take into consideration in the future? (No marks)

Since this document is written in English, we would appreciate if you could send your response in English too. We will still mark it and consider it if it is in another language, but this is our preference.

For the avoidance of any doubt, this document is copyrighted and shall not be re-distributed totally or partially without written consent.

Final notes:

Please, **do not get stressed or anxious when working on your submission**. This task is an opportunity for you to demonstrate what you know and are able to do. If you get stressed or stuck, please write to us as we might be able to help.

Based on the feedback we have received over time, it is better if you read the problem fully first, **think about it offline and try to resolve it away from a computer**. Starting to code without a clear plan, might result in you wasting a lot of time and getting frustrated.

You might have noticed that this task is about a specific area of programming, and you might think it is unfair, because there is more to development than this. If you thought that, you would be right, we know that. The reason we are asking you to complete this task is because we think it has some similarities to the problems that we face day-to-day in our job. We sometimes work on complex/abstract problems that require creative solutions. Our work is not just simple web development for a marketing site; normally we build web apps that do complex things such as processing data or analysis. Solving challenging problems like this is what we find exciting and fun, and we are hoping you enjoy these kinds of problems too.

Although there is a deadline for the submission of this task, if you feel that, for any reason, you need more time to complete this task; please, let us know at the earliest opportunity to discuss it with you.

Thank you very much and good luck! We are looking forward to seeing the solution you come up with.
We now ask you that you are a bit patient. Our ability to review your solutions really varies depending on how busy we are. Sometimes it is a matter of minutes, sometimes it is days. If you have sent us the email and you can see it in your sent folder, that should be enough. **We promise we will get back to you either way**, if it takes us longer than a week, please feel free to ask how we are doing with your return.

- Jose