



# C Piscine

## Rush01

*Summary: This document is the subject for Rush01 of the C Piscine @ 42.*

*Version: 7.4*

# Contents

<b>I</b>	<b>Foreword</b>	<b>2</b>
<b>II</b>	<b>Instructions</b>	<b>4</b>
<b>III</b>	<b>Subject</b>	<b>5</b>
<b>IV</b>	<b>Appendix</b>	<b>7</b>
IV.1	Appendix 1 . . . . .	7
IV.2	Appendix 2 . . . . .	7
IV.3	Appendix 3 . . . . .	8
IV.4	Appendix 4 . . . . .	9
<b>V</b>	<b>Submission and peer-evaluation</b>	<b>10</b>

# Chapter I

## Foreword

Here are some notable quotes from various movies:

1. "Find a truly original idea. It is the only way I will ever distinguish myself. It is the only way I will ever matter."

-A Beautiful Mind

2. "You don't have to be the bad guy. You are the most talented, most interesting, and most extraordinary person in the universe. And you are capable of amazing things. Because you are the Special. And so am I. And so is everyone. The prophecy is made up, but it's also true. It's about all of us. Right now, it's about you. And you... still... can change everything."

-The Lego Movie

3. "Sometimes it is the people who no one imagines anything of who do the things that no one can imagine." -The Imitation Game

4. "There should be no boundaries to human endeavour. We are all different. However bad life may seem, there is always something you can do, and succeed at. While there's life, there is hope."

-The Theory of Everything

5. "Just because someone stumbles and loses their path doesn't mean they're lost forever."

-X-Men Days of Future Past

6. "Where we're going, we don't need roads."

-Back to the Future

7. "I'm bad, and that's good. I will never be good, and that's not bad. There's no one I'd rather be than me."

-Wreck-it Ralph

8. "KA-ME-HA-ME-HAAAAAAAAAAAA"

-Various movies

While movie culture is important, it will not assist with this project.


# Chapter II

## Instructions

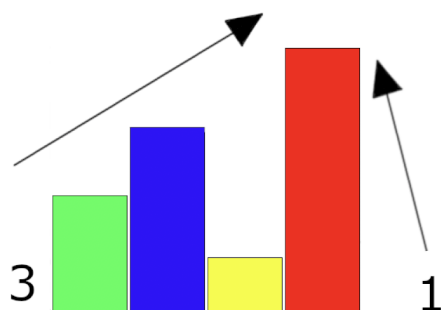
- You must do the project with the imposed team. Meeting with your teammates is part of your tasks, using any appropriate means (Slack, email, phone if available, or directly in person).
- If you have really tried everything to reach one of your teammates, but they still remain unreachable, follow the instructions from your pedagogical team, if provided. The default procedure is to do the project with the available teammates, and discuss the situation during the evaluation. Even if the group leader is missing, you still have access to the submission directory.
- Your work must comply with the Norm. If you have bonus files/functions, they are included in the norm check, and you will receive a score of 0 if there is a norm error.
- You must handle errors coherently. You may either print an error message or simply return control to the user.
- Your project must be completed and pushed to the Git repository by the deadline displayed on the project's main page on the intranet.
- You must follow the submission procedure described at the end of this document, if provided.
- Your program must compile using `cc` with the following flags: `-Wall -Wextra -Werror`. If there is a moulinette, it will use the same compiler and flags.
- If your program does not compile, you will receive a score of 0.
- The group will be automatically registered for the defense. You must attend your evaluation with all of your teammates. The purpose of the defense is to present and explain your work comprehensively.
- Do not cancel your evaluation; you will not get a second one.
- Each member of the group must be fully aware of all the details of the project. If you choose to split the workload, ensure you understand every part completed by other team members. This understanding may be verified during the evaluation.

# Chapter III

## Subject

	Exercise 00
Rush01	
Turn-in directory: <i>ex00/</i>	
Files to turn in: All the necessary files (*.c)	
Allowed functions: write, malloc, free	

- Your source code will be compiled as follows: `cc -Wall -Wextra -Werror -o rush01 *.c`.
- Your submission directory must contain all files required to compile your program.
- Create a program that solves the following problem:  
Given a 4x4 grid, place boxes of heights 1 to 4 on each available cell so that every row and column sees the correct number of boxes from each possible point of view (left/right for rows, top/bottom for columns).
- Example for one row or one column: The box of height 3 will hide the box of height 1 from the left, so there are 3 visible boxes. Only one box is visible from the right, as the box of height 4 hides everything.



- Each view (2 per row and 2 per column) will have a given value. Your program must place the boxes correctly, ensuring that each row and column contains only one box of each size.

- Your output must display the first solution you encounter.
- Here is how we will launch your program:

```
> ./rush01 "col1top col2top col3top col4top col1bottom col2bottom col3bottom col4bottom row1left  
row2left row3left row4left row1right row2right row3right row4right"
```

- "col1top" represents the value for the left column upper point of view, etc. Refer to appendix 1 to see what represents each element.
- Each element of the string is a number ranging between '1' and '4'.
- This is the only acceptable input for your program. Any other input must be considered an error.
- Here is an example of intended input/output for a valid set.

```
./rush01 "4 3 2 1 1 2 2 2 4 3 2 1 1 2 2 2" | cat -e  
1 2 3 4$  
2 3 4 1$  
3 4 1 2$  
4 1 2 3$
```

- Refer to appendix 2 and 3 for a flat vision, and appendix 4 for a 3D vision.
- In case of an error or if you cannot find any solutions, display "Error" followed by a newline.
- If you want bonus points, you may try to handle other map sizes (up to 9x9).
- As usual, if a bonus works but the mandatory one fails the tests, you will receive a score of 0.

# Chapter IV

## Appendix

What follows is an artistic view of your program. You need to submit a program as described in the previous chapter.

The sole purpose of these representations is to help you understand the project.

### IV.1 Appendix 1

	col1top	col2top	col3top	col4top	
row1left					row1right
row2left					row2right
row3left					row3right
row4left					row4right
	col1bottom	col2bottom	col3bottom	col4bottom	

Representation of the different points of vision: colXtop, colXbottom, rowXleft and rowXright.

### IV.2 Appendix 2

	4	3	2	1	
4					1
3					2
2					2
1					2
	1	2	2	2	

With assigned values for col\* and row\*, we get this.

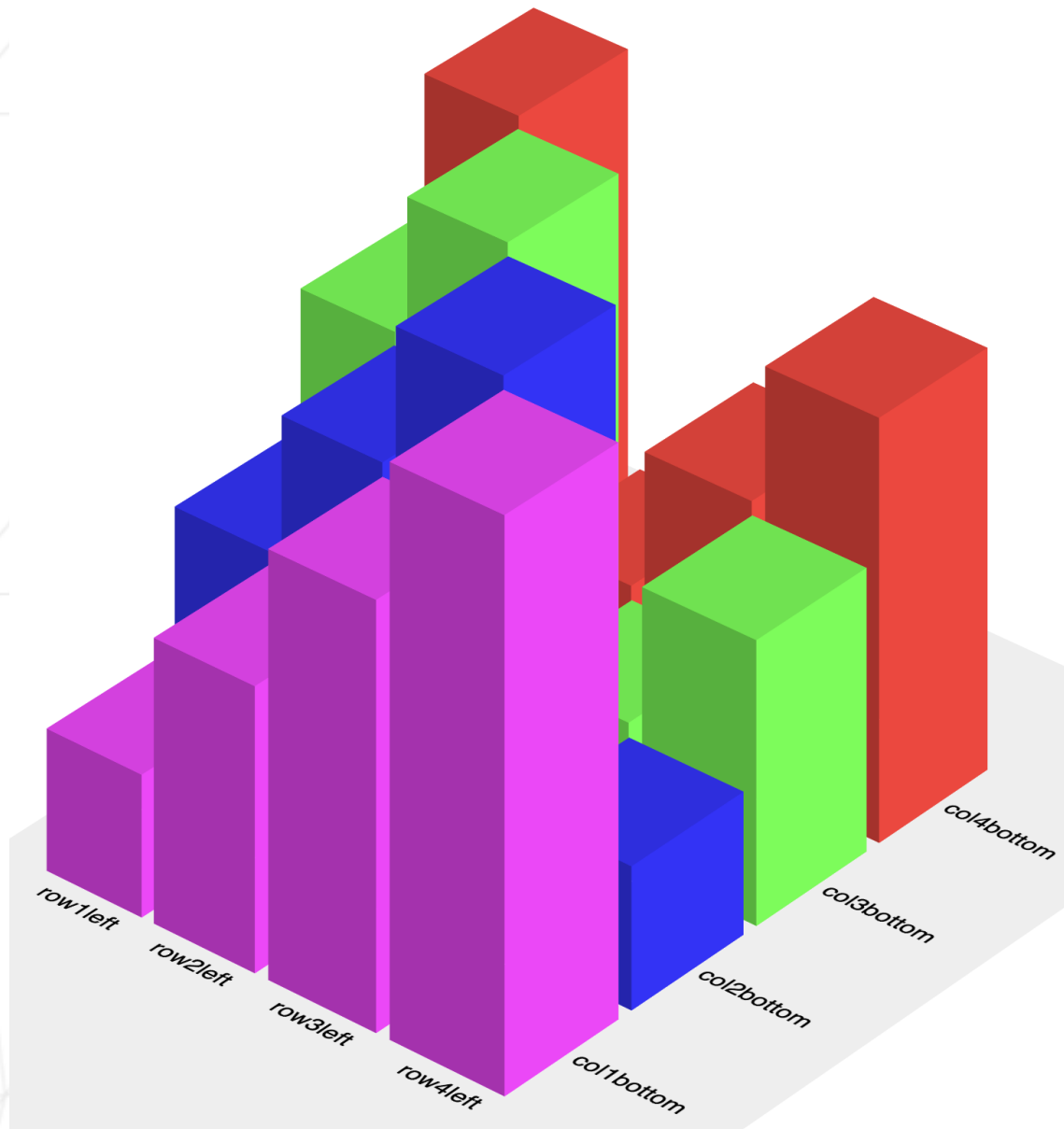


## IV.3 Appendix 3

	4	3	2	1	
4	1	2	3	4	1
3	2	3	4	1	2
2	3	4	1	2	2
1	4	1	2	3	2
	1	2	2	2	

Your program must fill in the blanks of the 4x4 grid using the rules described above in this document.

## IV.4 Appendix 4



A 3D view of the 4x4 grid and the blocs.

# Chapter V

## Submission and peer-evaluation

Submit your assignment to your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Do not hesitate to double-check the names of your files to ensure they are correct.

This assignment is not verified by a program. You are free to organize your files as you wish, provided you submit the mandatory files and comply with the requirements.



Submit only the files requested by the subject of this project.