



# C Bootcamp

## 24h Challenge

Staff WeThinkCode\_ [staff@wethinkcode.co.za](mailto:staff@wethinkcode.co.za)

*Summary: THE FOLLOWING TAKES PLACE BETWEEN 11.00 P.M. AND 12.00 A.M.*

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>4</b>
<b>III</b>	<b>ft_spy.c</b>	<b>5</b>

# Chapter I

## Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. So if you want to avoid bad surprises, be as thorough as possible.
- Moulinette is not very open-minded. It won't try and understand your code if it doesn't respect the Norm. Moulinette relies on a program called **Norminator** to check if your files respect the norm. TL;DR: it would be idiotic to submit a piece of work that doesn't pass **Norminator**'s check.
- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We **will not** take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- If `ft_putchar()` is an authorized function, we will compile your code with our `ft_putchar.c`.
- You'll only have to submit a `main()` function if we ask for a program.

- Moulinette compiles with these flags: -Wall -Wextra -Werror, and uses gcc.
- If your program doesn't compile, you'll get 0.
- You cannot leave any additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet / ....`
- Check out the "C Bootcamp" part of the forum on the intranet.
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- By Odin, by Thor ! Use your brain !!!

# Chapter II

## Foreword

« CTU's reception desk, the Counter Terrorist Unit, the most secured building in New York, what can I do for you??

- We are the cleaning team. »

M. Boulon, the building's night security guard, raised his eyebrows. There was something fishy with the team in front of him. For example, those clothes were too tight to allow movement like hovering. But it was 2h15AM who was he to discuss the direction's orders?

«Access granted.

- We'll take the lift. We'll first check it.

- Yes, yes no need to give me your planning, I have some other people waiting. Mister, you are?

- Wanted by 2 2 international superpower as well as the first MacDonald's exportator country. I need to go in the CTU's surveillance room to prevent a terrorist attack. I am... Nick Bauer. »

Again, it was 2h15AM.

« Access... granted. Good luck? »

As soon as arrived in the surveillance room, Nick decided to go ahead: only his mission mattered.


« All of you! I don't have time to explain who I am, facebook me if you're new. I'm now in charge of this unit to prevent the Operation Twilight! You, there! The innocent intern with a coffee! It's your lucky day: you're promoted to Special Agent. What's your name??

- My name is... - Whatever, I'll call you Pancake from now on. Special Agent Pancake, we've got a terrorist network to dismantle! »

# Chapter III

## ft\_spy.c

Help the Special Agent Pancake to analyse the enemy communication with the next exercise.

	Exercise 08
ft_spy.c	
Turn-in directory : <i>ex08/</i>	
Files to turn in : <b>ft_spy.c</b>	
Allowed functions : <b>write</b>	
Notes : <b>n/a</b>	

- Create the file `ft_spy.c` which displays **Alert!!!** on the standard output, followed by a line break, if any of the given arguments is one of these words: **president**, **attack** or **Powers**.
- Evidently, your `ft_spy.c` file should have a **main**.

```
$> ./ft_spy "I" "will" "kill" "Austin" "Powers"
Alert!!!
$> ./ft_spy "Hello" "I" "want" "a" "4" "cheese" "pizza"
$> ./ft_spy "The attack" "will" "start" "soon"
$> ./ft_spy "an" "attack" "will" "start" "soon"
Alert!!!
$>
```

- Examine the examples thoroughly :

```
$> ./ft_spy "I" "will" "kill" "Nigel" "PoWeRs"
Alert!!!
$> ./ft_spy " the " " president " "must" "die "
Alert!!!
$> ./ft_spy " the'" "omnipresident" "must" "eat "
$>
```



Why not take advantage of this exercise to add reuseable functions to your lib ?