C Bootcamp

24h Challenge

Staff WeThinkCode_ staff@wethinkcode.co.za

Summary: THE FOLLOWING TAKES PLACE BETWEEN 8.00 P.M. AND 9.00 P.M.

Contents

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

« We did it! We can now go look for the antidote... »

Vic's happyness was only of short duration. The neurotoxins have had more time to attack her brain than Nick's and already the first sign of the infection showed.

« Keep up babe, screamed Nick, be brave! We're gonna make it together! The Windows 8 tablet don't have that much battery anymore but enough to allow us to read the decoded message. »

$OPERATION TWILIGHT
$WHAT TO DO IF YOU GOT INFECTED IN THE JERSEY SHORE HOUSE
$Agent, don't panic.
$An antidote is available in the bathroom's first-aid cabinet.

« God Vic, faster, run in that damn bathroom!

$For security reasons, the antidote is hidden in a vial tagged H2SO4

« Be careful with that vial, it burns, it's acid!»

$The first rigged vial is hearvier than the antidote.
$The second rigged vial is lighter than the antidote.
$The antidote's effect is instant.
$It's not advised to ingest another vial than the antidote itself.

« Nick, we must give up and accept that we'll die!

- I am Nick Bauer, Vic. I'll find the antidote. The Operation Twilight's terrorists already are one hour ahead of us, but I'll just need another couple of minutes to find the solution of this evil puzzle and get revange. »

Chapter III

ft_button.c

If you're as skilled as Nick Bauer, you should be able to write a program that will allow you to find out which vial is the antidote.
HINT : it's the one in the middle.

	Exercise 05
ft_button.c	
Turn-in directory : *ex*05/	
Files to turn in : `ft_button.c`	
Allowed functions : `None`	
Notes : `n/a`	

- Create a function `ft_button` which takes three `ints` as argument and returns the middle value.

- Here's how it should be prototyped :

```
int ft_button(int i, int j, int k);
```

To know more about the middle value, read the foreword section...