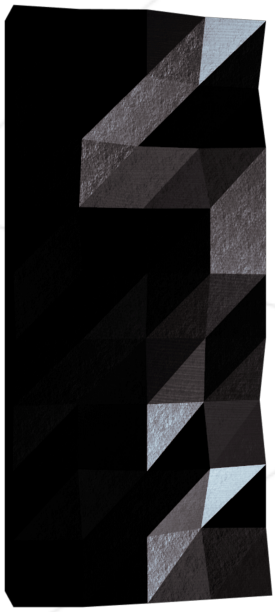

C Bootcamp

Day 24
The Spy Who Coded me

Staff WeThinkCode_ info@wethinkcode.co.za

Summary: *THE FOLLOWING TAKES PLACE BETWEEN 9.00 A.M. AND 10.00 A.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

Vanessa Kensington

For a long time nobody reactivated my processor. Of course, I've been destroyed in an explosion, but my software has been downloaded in another model. I remained hibernating all this time. 12 hours ago, someone reactivated me. My meeting with Austin made me a different female-bot, capable of its own actions. When all my sisters destroyed the factory that we were in, I preferred to follow my instincts. I love Austin, I always did.

I have been looking for him on the Internet. I infiltrated without shuddering the secret headquarters of Dr. Evil's network. I got his plans and went to the President to protect him.

But I failed.

When Dr. Evil came before me, his « laser » gun in one hand and a jar of marmite half opened in the other, I lost my composure. These unbearable images reminded me of a moment of weakness : one where, exploding, I almost took Austin's life many years ago. Why must a female-bot always behave like a women-bot ?

« Never again », I thought when falling on my knees. I would never hurt anyone again just because I've been coded this way. I felt my synthetic synapses failed. I had to obey Dr. Evil and hurt Austin. I resisted.

But at what cost ?

Too many times, we ask ourselves if androids dream of Electric Sheep. But who knows that female-bots can actually cry ?

I would like to see Austin Powers one last time ...

Chapter III

ft_join.c

`Grant Vanessa Kensington's last wish by allowing Austin to reunite with her.`

	Exercise 18
ft_join.c	
Turn-in directory : *ex*18/	
Files to turn in : `ft_join.c`	
Allowed functions : `malloc`	
Notes : `n/a`	

- Create a function `ft_join` that merges the array given as a parameter into one single character string. The separation argument will serve as "glue" between elements.
- Here's how it should be prototyped :

```
char  *ft_join(char **tab, char *sep);
```

`Smells like a reuseable function, does it not ?`

`You should know how arrays end when coded properly...`