

C Bootcamp

Day 08

 $Staff\ We Think Code_\ {\tt info@wethinkcode.co.za}$

Summary: This document is the subject for Day08 of the C Bootcamp @ $WeThinkCode_{_}$.

Contents

Ι	Instructions	2
II	Foreword	4
III	Exercise 00: ft.h	5
IV	Exercise 01 : ft_boolean.h	6
V	Exercise 02 : ft_abs.h	8
VI	Exercise 03 : ft_point.h	9
VII	Exercise 04 : ft_param_to_tab	10
VIII	Exercise 05 : ft_show_tab	12

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.

C Bootcamp

Day 08

- Moulinette compiles with these flags: -Wall -Wextra -Werror, and uses gcc.
- If your program doesn't compile, you'll get 0.
- ullet You <u>cannot</u> leave <u>any</u> 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

Here's how to connect an XBOX ONE:

- 1. Connect the power adapter into a plug, depending on the country, you might need an adapter.
- 2. Connect the HDMI connector to the HDMI port on the console.
- 3. Connect the HDMI connector on the HDMI port on your huge HDTV or monitor.

optional:

4. If you have a great 7.1 sound system use the optical cable to connect it. In some very specific cases, a 5.1 sound system will also be accepted.

Of course to connect a PS4 it's much more simpler!

Chapter III

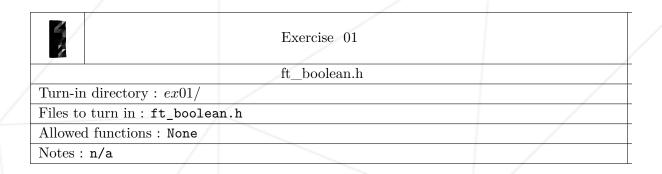
Exercise 00: ft.h

	Exercise 00	
/	${ m ft.h}$	/
Turn-in directory : $ex00/$		
Files to turn in: ft.h		
Allowed functions : None		
Notes: n/a		

- Create your ft.h file.
- It contains all prototypes of your libft.a functions.

Chapter IV

Exercise 01: ft_boolean.h



• Create a ft_boolean.h file. It'll compile and run the following main appropriately :

• This program should display

I have an even number of arguments.

Chapter V

Exercise 02 : ft_abs.h

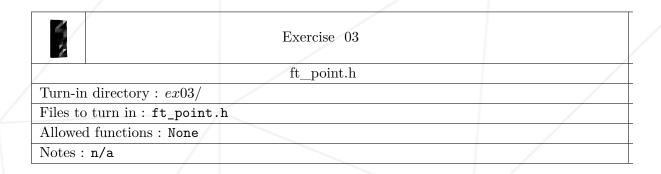
	Exercise 02	
/	ft_abs.h	
Turn-in directory : $ex02/$		
Files to turn in : ft_abs.h		
Allowed functions: None		
Notes : n/a		

 \bullet Create a macro ABS which replaces its argument by it absolute value :

#define ABS(Value)

Chapter VI

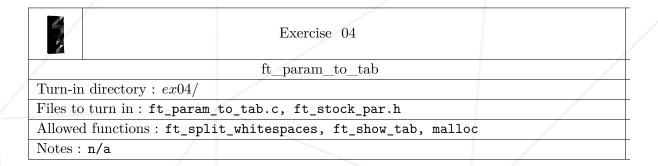
Exercise 03: ft_point.h



• Create a file ft_point.h that'll compile the following main :

Chapter VII

Exercise 04: ft_param_to_tab



- Create a function that stores the program's arguments within an array structure and that returns the address of that array's first box.
- All elements of the array must be processed, including av [0].
- Here's how it should be prototyped:

```
struct s_stock_par *ft_param_to_tab(int ac, char **av);
```

• The structure array should be allocated and its last box shall contain 0 in its str element to point out the end of the array.

C Bootcamp Day 08

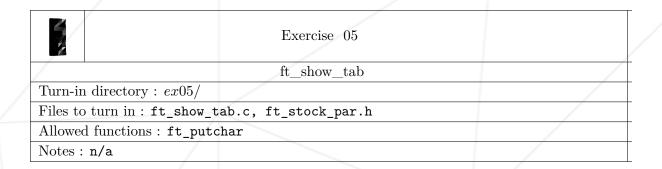
• The structure is defined in the ft_stock_par.h file, like this:

```
typedef struct s_stock_par
{
   int size_param;
   char *str;
   char *copy;
   char **tab;
}
   t_stock_par;
```

- size_param being the length of the argument;
- \circ str being the address of the argument ;
- copy being the copy of the argument;
- tab being the array returned by ft_split_whitespaces.
- We'll test your function with our ft_split_whitespaces and our ft_show_tab (next exercise). Take the appropriate measures for this to work!

Chapter VIII

Exercise 05: ft_show_tab



- Create a function that displays the content of the array created by the previous function.
- Here's how it should be prototyped :

```
void ft_show_tab(struct s_stock_par *par);
```

• The structure is defined in the ft_stock_par.h file, like this:

```
typedef struct s_stock_par
{
  int size_param;
  char *str;
  char *copy;
  char **tab;
}
  t_stock_par;
```

- For each box, we'll display (one element per line):
 - \circ the argument
 - the size
 - each word (one per line)