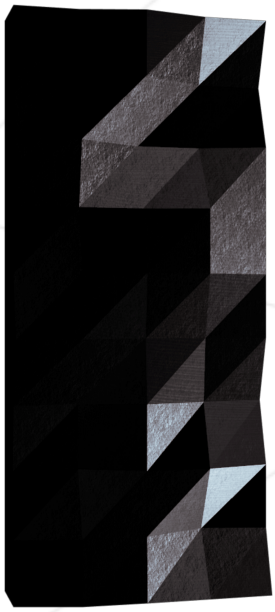

C Bootcamp

Day 02

Staff WeThinkCode_ info@wethinkcode.co.za

Summary: This document is the subject for Day02 of the C Bootcamp @ WeThinkCode_.

Contents

I	Instructions	2
II	Foreword	4
III	Exercise 00 : ft_print_alphabet	6
IV	Exercise 01 : ft_print_reverse_alphabet	7
V	Exercise 02 : ft_print_numbers	8
VI	Exercise 03: ft_is_negative	9
VII	Exercise 04 : ft_print_comb	10
VIII	Exercise 05 : ft_print_comb2	11
IX	Exercise 06 : ft_putnbr	12
X	Exercise 07 : ft_print_combn	13

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 all your exercises.
- 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 !!!

`Norminator must be launched with the -R CheckForbiddenSourceHeader flag. Moulinette will use it too.`

Chapter II

Foreword

William Banting was a British undertaker who was very obese and desperately wanted to lose weight. In the year 1862 he paid a visit to his doctor, William Harvey, who proposed a radical eating plan that was high in fat but included very few carbohydrates. By following this eating plan Banting experienced such remarkable weight loss that he wrote an open letter to the public, the "Letter on Corpulence", which became widely distributed. As more people started following this eating plan to lose weight, the term "banting" or to "bant" became popularised.

Banting merely discovered what human beings were designed to eat: what early humans ate 200,000 years ago. Respected biologists, geneticists, paleoanthropologists and theorists believe that human genes have hardly changed since human beings began their journey on earth. If you could put the entire human history into one day, we have only been eating cereals and grains for five minutes and sugar for five seconds, a very short amount of time in our existence. After the success experienced by William Banting on this low-carb, high-fat eating plan, the "banting" diet became the standard treatment for weight loss in all major European and North American medical schools. But in 1959 it was excluded from all the major medical and nutritional textbooks.

In 1977 the US government published the Dietary Goals for the United States, a set of guidelines that advocated a diet high in carbs and low in fat, exactly the opposite of the diet we have been following for much of our existence. It was decreed that we should eat six to eleven portions of grains per day and that sugar was absolutely fine to add to everything. This diet was subsequently adopted across most of the Western world and a plethora of low fat-food products hit the shelves. This has had a disastrous effect on our health. Since the early 1980's the incidence of obesity and diabetes has risen rapidly. Can we really call this a coincidence?

There is a common misconception that eating fat, especially saturated fat, is bad for you and that it is a primary cause of high blood pressure, heart disease and obesity. This is simply not true and was based on a flawed study by Ancel Keys in 1953. The truth is that a diet high in carbohydrates, particularly refined carbohydrates and sugar are the cause of obesity, diabetes as well as other chronic illnesses. Vegetable (seed) oils and their derivatives such as margarine are also a contributing factor to heart disease, although manufacturers tell us the exact opposite.

This might come as a surprise, but of the three macronutrients in our diet (protein, fat and carbohydrates), only carbohydrates are non-essential for human life. We cannot function properly for more than a few days without eating fat; without an adequate protein intake we develop protein-calorie malnutrition within a few months. But avoiding carbohydrate has no short- or long-term effects on humans, other than the (usually beneficial) effect of weight loss, especially in those who are the most overweight. While we need a constant supply of glucose, it can be produced by the liver from fat and protein and doesn't need to be ingested as carbohydrate in our diets.

The usual refrain of anyone looking at banting for the first time is "but what about my cholesterol?" There is much evidence to support the fact that cholesterol is not the culprit in heart disease. A bit like a policeman being at the scene of the crime being blamed for the crime - cholesterol will only adhere to a 'leaking' artery wall which is damaged by inflammation - to protect you. By living on carbs and sugar those arteries remain inflamed. Sugar is the most inflammatory thing you can put into your mouth, and will continue to rob you of perfect health. Grains are turned into sugar by the body. So a high carbohydrate diet will always foster inflammation in the body, not only in the arteries but the brain, liver, digestive tract and joints leading to many of the chronic diseases we see today which are supposedly 'incurable'. Many people report relief from all the above in a relatively short time after adopting the Banting lifestyle.

There are different views on Banting. But C is always good, let's stick to that!

Chapter III

Exercise 00 : ft_print_alphabet

Exercise 00

ft_print_alphabet
Turn-in directory : $ex00/$
Files to turn in : `ft_print_alphabet.c`
Allowed functions : `ft_putchar`
Notes : `n/a`

- Create a function that displays the alphabet in lowercase, on a single line, by ascending order, starting from the letter 'a'.
- Here's how it should be prototyped :

```
void ft_print_alphabet(void);
```

Chapter IV

Exercise 01 : ft_print_reverse_alphabet

	Exercise 01
	ft_print_reverse_alphabet
Turn-in directory : *ex*01/	
Files to turn in : `ft_print_reverse_alphabet.c`	
Allowed functions : `ft_putchar`	
Notes : `n/a`	

- Create a function that displays the alphabet in lowercase, on a single line, by descending order, starting from the letter 'z'.

sddasd

- Here's how it should be prototyped :

```
void ft_print_reverse_alphabet(void);
```

Chapter V

Exercise 02 : ft_print_numbers

Exercise 02

ft_print_numbers
Turn-in directory : *ex*02/
Files to turn in : `ft_print_numbers.c`
Allowed functions : `ft_putchar`
Notes : `n/a`

- Create a function that displays all digits, on a single line, by ascending order.
- Here's how it should be prototyped :

```
void ft_print_numbers(void);
```

Chapter VI

Exercise 03: ft_is_negative

	Exercise 03
	ft_is_negative
Turn-in directory : *ex*03/	
Files to turn in : `ft_is_negative.c`	
Allowed functions : `ft_putchar`	
Notes : `n/a`	

- Create a function that displays 'N' or 'P' depending on the integer's sign entered as a parameter. If `n` is negative, display 'N'. If `n` is positive or null, display 'P'.

- Here's how it should be prototyped :

```
void ft_is_negative(int n);
```

Chapter VII

Exercise 04 : ft_print_comb

Exercise 04
ft_print_comb
Turn-in directory : *ex*04/
Files to turn in : `ft_print_comb.c`
Allowed functions : `ft_putchar`
Notes : `n/a`

- Create a function that displays all different combinations of three different digits in ascending order, listed by ascending order - yes, repetition is voluntary.
- Here's the intended output :

```
$>./a.out | cat -e
012, 013, 014, 015, 016, 017, 018, 019, 023, ..., 789$>
```

- 987 isn't there because 789 already is.
- 999 isn't there because the digit 9 is present more than once.
- Here's how it should be prototyped :

```
void ft_print_comb(void);
```

Chapter VIII

Exercise 05 : ft_print_comb2

Exercise 05

ft_print_comb2

Turn-in directory : *ex*05/

Files to turn in : `ft_print_comb2.c`

Allowed functions : `ft_putchar`

Notes : `n/a`

- Create a function that displays all different combination of two digits between 00 and 99, listed by ascending order.

- Here's the expected output :

```
$>./a.out | cat -e
00 01, 00 02, 00 03, 00 04, 00 05, ..., 00 99, 01 02, ..., 97 99, 98 99$>
```

- Here's how it should be prototyped :

```
void ft_print_comb2(void);
```

Chapter IX

Exercise 06 : ft_putnbr

	Exercise 06
	ft_putnbr
Turn-in directory : *ex*06/	
Files to turn in : `ft_putnbr.c`	
Allowed functions : `ft_putchar`	
Notes : `n/a`	

- Create a function that displays the number entered as a parameter. The function has to be able to display all possible values within an `int` type variable.
- Here's how it should be prototyped :

```
void ft_putnbr(int nb);
```

- For example:
  - `ft_putnbr(42)` displays "42".

Chapter X

Exercise 07 : ft_print_combn

Exercise 07
ft_print_combn
Turn-in directory : *ex*07/
Files to turn in : `ft_print_combn.c`
Allowed functions : `ft_putchar`
Notes : `n/a`

- Create a function that displays all different combinations of `n` numbers by ascending order.
- n will be so that : 0 < n < 10.
- If `n` = 2, here's the expected output :

```
$>./a.out | cat -e
01, 02, 03, ..., 09, 12, ..., 79, 89$>
```

- Here's how it should be prototyped :

```
void ft_print_combn(int n);
```