Contents

- 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 Piscine" 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

Cod liver oil is a nutritional supplement derived from liver of cod fish (Gadidae).

As with most fish oils, it has high levels of the omega-3 fatty acids, eicosapentaenoic acid (EPA) and docosahexaenoic acid (DHA).
Cod liver oil also contains vitamin A and vitamin D.

It has historically been taken because of its vitamin A and vitamin D content.

It was once commonly given to children, because vitamin D has been shown to prevent rickets and other symptoms of vitamin D deficiency.

Contrary to Cod liver oil, C is good, eat some!

Chapter III

Exercise 00 : ft_print_alphabet

	Exercice : 00
ft_print_alphabet	
Turn-in directory : $ex00/$	
Files to turn in : `ft_print_alphabet.c`	
Allowed functions : `ft_putchar`	
Remarks : `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

	Exercice : 01
ft_print_reverse_alphabet	
Turn-in directory : $ex01/$	
Files to turn in : `ft_print_reverse_alphabet.c`	
Allowed functions : `ft_putchar`	
Remarks : `n/a`	

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

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

Chapter V

Exercise 02 : ft_print_numbers

	Exercice : 02
ft_print_numbers	
Turn-in directory : $ex02/$	
Files to turn in : `ft_print_numbers.c`	
Allowed functions : `ft_putchar`	
Remarks : `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

	Exercice : 03
ft_is_negative	
Turn-in directory : $ex03/$	
Files to turn in : `ft_is_negative.c`	
Allowed functions : `ft_putchar`	
Remarks : `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

	Exercice : 04
ft_print_comb	
Turn-in directory : $ex04/$	
Files to turn in : `ft_print_comb.c`	
Allowed functions : `ft_putchar`	
Remarks : `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

	Exercice : 05
ft_print_comb2	
Turn-in directory : $ex05/$	
Files to turn in : `ft_print_comb2.c`	
Allowed functions : `ft_putchar`	
Remarks : `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 X

Exercise 07 : ft_print_combn

	Exercice : 07
ft_print_combn	
Turn-in directory : $ex07/$	
Files to turn in : `ft_print_combn.c`	
Allowed functions : `ft_putchar`	
Remarks : `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);
```