Chapter III

Exercise 00 : ft_ft

	Exercice : 00
ft_ft	
Turn-in directory : *ex*00/	
Files to turn in : `ft_ft.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function that takes a pointer to int as a parameter, and sets the value "42" to that int.
- Here's how it should be prototyped :

```
void        ft_ft(int *nbr);
```

Chapter IV

Exercise 01 : ft_ultimate_ft

	Exercice : 01
ft_ultimate_ft	
Turn-in directory : $ex01/$	
Files to turn in : `ft_ultimate_ft.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function that takes a pointer to pointer to pointer to pointer to pointer to pointer to pointer to pointer to pointer to int as a parameter and sets the value "42" to that int.

- Here's how it should be prototyped :

```
void        ft_ultimate_ft(int *********nbr);
```

Chapter V

Exercise 02 : ft_swap

	Exercice : 02
ft_swap	
Turn-in directory : $ex02/$	
Files to turn in : `ft_swap.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function that swaps the value of two integers whose addresses are entered as parameters.
- Here's how it should be prototyped :

```
void    ft_swap(int *a, int *b);
```

Chapter VI

Exercise 03 : ft_div_mod

	Exercice : 03
ft_div_mod	
Turn-in directory : $ex03/$	
Files to turn in : `ft_div_mod.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function `ft_div_mod` prototyped like this :

```
void    ft_div_mod(int a, int b, int *div, int *mod);
```

- This function divides parameters `a` by `b` and stores the result in the int pointed by `div`. It also stores the remainder of the division of `a` by `b` in the int pointed by `mod`.

Chapter VII

Exercise 04 : ft_ultimate_div_mod

	Exercice : 04
ft_ultimate_div_mod	
Turn-in directory : $ex04/$	
Files to turn in : `ft_ultimate_div_mod.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function `ft_ultimate_div_mod` with the following prototype :

```
void    ft_ultimate_div_mod(int *a, int *b);
```

- This function divides parameters `a` by `b`. The result of this division is stored in the int pointed by `a`. The remainder of the division is stored in the int pointed by `b`.

Chapter VIII

Exercise 05 : ft_putstr

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

- Create a function that displays a string of characters on the standard output.
- Here's how it should be prototyped :

```
void    ft_putstr(char *str);
```

Chapter IX

Exercise 06 : ft_strlen

	Exercice : 06
ft_strlen	
Turn-in directory : $ex06/$	
Files to turn in : `ft_strlen.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function that counts and returns the number of characters in a string.
- Here's how it should be prototyped :

```
int        ft_strlen(char *str);
```

Chapter X

Exercise 07 : ft_strrev

	Exercice : 07
ft_strrev	
Turn-in directory : *ex*07/	
Files to turn in : `ft_strrev.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function that reverses the order of characters in a string.
- It has to return str.
- Here's how it should be prototyped :

```
char        *ft_strrev(char *str);
```

- Example:

```
a => a
ab => ba
abcde => edcba
```

Chapter XII

Exercise 09 : ft_sort_integer_table

	Exercice : 09
ft_sort_integer_table	
Turn-in directory : $ex09/$	
Files to turn in : `ft_sort_integer_table.c`	
Allowed functions : `Nothing`	
Remarks : `n/a`	

- Create a function which sorts an array (table) of integers by ascending order.
- The arguments are a pointer to int and the number of ints in the array.
- Here's how it should be prototyped :

```
void    ft_sort_integer_table(int *tab, int size);
```