


Chapitre III

Exercice 00 : ft_putchar

	Exercice : 00
	ft_putchar
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : ft_putchar.c	
Fonctions Autorisées : write	

- Écrire une fonction qui affiche le caractère passé en paramètre.
- Elle devra être prototypée de la façon suivante :


```
void ft_putchar(char c);
```

Pour afficher le caractère, vous devez utiliser la fonction `write` de la manière suivante.

```
write(1, &c, 1);
```

Chapitre IV

Exercice 01 : ft_print_alphabet


	Exercice : 01
	ft_print_alphabet
	Dossier de rendu : ex01/
	Fichiers à rendre : ft_print_alphabet.c
	Fonctions Autorisées : write

- Écrire une fonction qui affiche l'alphabet en minuscule sur une seule ligne, dans l'ordre croissant, à partir de la lettre 'a'.
- Elle devra être prototypée de la façon suivante :

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

Chapitre V

Exercice 02 : ft_print_reverse_alphabet


	Exercice : 02
	ft_print_reverse_alphabet
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <code>ft_print_reverse_alphabet.c</code>	
Fonctions Autorisées : <code>write</code>	

- Écrire une fonction qui affiche l'alphabet en minuscule sur une seule ligne, dans l'ordre décroissant, à partir de la lettre 'z'.
- Elle devra être prototypée de la façon suivante :

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

Chapitre VI

Exercice 03 : ft_print_numbers


	Exercice : 03
ft_print_numbers	
Dossier de rendu : ex03/	
Fichiers à rendre : ft_print_numbers.c	
Fonctions Autorisées : write	

- Écrire une fonction qui affiche tous les chiffres sur une seule ligne, dans l'ordre croissant.
- Elle devra être prototypée de la façon suivante :

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

Chapitre VII

Exercice 04 : ft_is_negative


	Exercice : 04
	ft_is_negative
	Dossier de rendu : ex04/
	Fichiers à rendre : ft_is_negative.c
	Fonctions Autorisées : write

- Écrire une fonction qui affiche 'N' ou 'P' suivant le signe de l'entier passé en paramètre. Si `n` est négatif alors afficher 'N'. Si `n` est positif ou nul alors afficher 'P'.
- Elle devra être prototypée de la façon suivante :

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

Chapitre VIII

Exercice 05 : ft_print_comb

	Exercice : 05
	ft_print_comb
	Dossier de rendu : <i>ex05/</i>
	Fichiers à rendre : <code>ft_print_comb.c</code>
	Fonctions Autorisées : <code>write</code>

- Écrire une fonction qui affiche, dans l'ordre croissant, toutes les différentes combinaisons de trois chiffres différents dans l'ordre croissant - oui, la répétition est volontaire.
- Cela donne quelque chose comme ça :


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

- 987 n'est pas là car 789 est déjà présent
- 999 n'est pas là car ce nombre ne comporte pas exclusivement des chiffres différents les uns des autres
- Elle devra être prototypée de la façon suivante :

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

Chapitre IX

Exercice 06 : ft_print_comb2

	Exercice : 06
	ft_print_comb2
	Dossier de rendu : <i>ex06/</i>
	Fichiers à rendre : <code>ft_print_comb2.c</code>
	Fonctions Autorisées : <code>write</code>

- Écrire une fonction qui affiche toutes les différentes combinaisons de deux nombres entre 0 et 99, dans l'ordre croissant.
- Cela donne quelque chose comme ça :


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

- Elle devra être prototypée de la façon suivante :

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

Chapitre X

Exercice 07 : ft_putnbr

	Exercice : 07
	ft_putnbr
Dossier de rendu : <i>ex07/</i>	
Fichiers à rendre : <code>ft_putnbr.c</code>	
Fonctions Autorisées : <code>write</code>	


- Écrire une fonction qui affiche un nombre passé en paramètre. La fonction devra être capable d'afficher la totalité des valeurs possibles dans une variable de type `int`.
- Elle devra être prototypée de la façon suivante :

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

- Par exemple :
 - `ft_putnbr(42)` affiche "42".

Chapitre XI

Exercice 08 : ft_print_combn

	Exercice : 08
	ft_print_combn
	Dossier de rendu : ex08/
	Fichiers à rendre : ft_print_combn.c
	Fonctions Autorisées : write

- Écrire une fonction qui affiche toutes les différentes combinaisons de n chiffres dans l'ordre croissant.
- n sera tel que : $0 < n < 10$.
- Si $n = 2$, cela donne quelque chose comme ça :

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

- Elle devra être prototypée de la façon suivante :

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