

C Piscine

Exercices C Basiques

Summary: This document is a compilation d'exercices C basiques for the C Piscine @ 42.

Contents

I	Instructions	2
II	Exercise 00 : ft_putchar	3
III	Exercise 01 : ft_print_alphabet	4
IV	Exercise 02 : ft_is_negative	5
V	Exercise 03 : ft_iterative_power	6
VI	Exercise 04 : ft_recursive_power	7
VII	Exercise 05 : ft_putnbr	8
VIII	Exercise 06 : ft_putstr	9
IX	Exercise 07 : ft_strlen	10
X	Exercise 08 : ft_str_is_alpha	11
XI	Exercise 09 : ft_str_is_printable	12
XII	Exercise 10 : ft_strlowercase	13
XIII	Exercise 11 : ft_strcpy	14
XIV	Exercise 12 : ft_ft	15
XV	Exercise 13 : ft_swap	16
XVI	Exercise 14 : ft_foreach	17
XVII	Exercise 15 : ft_print_param	18
XVIII	Exercise 16 : ft_main_printf	19

Chapter I

Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change up 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.
- 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.
- You'll only have to submit a `main()` function if we ask for a program.
- 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 /
- 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 !!!

The Norminette is mandatory unless stated otherwise on the exercise.

Chapter II

Exercise 00 : ft_putchar

	Exercise 00
	ft_putchar
	Turn-in directory : ex00/
	Files to turn in : ft_putchar.c

- Create a function that displays the character passed as a parameter.
- Here's how it should be prototyped :

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

Chapter III

Exercise 01 : ft_print_alphabet

	Exercise 01
	ft_print_alphabet
	Turn-in directory : ex01/
	Files to turn in : ft_print_alphabet.c

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

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

Chapter IV

Exercise 02 : ft_is_negative

	Exercise 02
	ft_is_negative
	Turn-in directory : ex02/
	Files to turn in : ft_is_negative.c

- Create a function that displays 'N' if the integer passed as a parameter is negative, or 'P' if it is positive or null.
- Here's how it should be prototyped :

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

Chapter V

Exercise 03 : ft_iterative_power

	Exercise 03
	ft_iterative_power
	Turn-in directory : ex03/
	Files to turn in : ft_iterative_power.c

- Create a function that returns the value of a number raised to a power using an iterative method.
- Here's how it should be prototyped :

```
int ft_iterative_power(int nb, int power);
```

Chapter VI

Exercise 04 : ft_recursive_power

	Exercise 04
	ft_recursive_power
	Turn-in directory : ex04/
	Files to turn in : ft_recursive_power.c

- Create a function that returns the value of a number raised to a power using a recursive method.
- Here's how it should be prototyped :

```
int ft_recursive_power(int nb, int power);
```

Chapter VII

Exercise 05 : ft_putnbr

	Exercise 05
	ft_putnbr
	Turn-in directory : ex05/
	Files to turn in : ft_putnbr.c

- Create a function that displays an integer.
- Here's how it should be prototyped :

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

Chapter VIII

Exercise 06 : ft_putstr

	Exercise 06
	ft_putstr
	Turn-in directory : ex06/
	Files to turn in : ft_putstr.c

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

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

Chapter IX

Exercise 07 : ft_strlen

	Exercise 07
	ft_strlen
	Turn-in directory : ex07/
	Files to turn in : ft_strlen.c

- Create a function that returns the length of a string.
- Here's how it should be prototyped :

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

Chapter X

Exercise 08 : ft_str_is_alpha

	Exercise 08
	ft_str_is_alpha
	Turn-in directory : ex08/
	Files to turn in : ft_str_is_alpha.c

- Create a function that returns 1 if the string given as a parameter contains only alphabetical characters, and 0 if it contains any other character.
- Here's how it should be prototyped :

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

- It should return 1 if str is empty.

Chapter XI

Exercise 09 : ft_str_is_printable

	Exercise 09
	ft_str_is_printable
	Turn-in directory : ex09/
	Files to turn in : ft_str_is_printable.c

- Create a function that returns 1 if the string given as a parameter contains only printable characters, and 0 if it contains any other character.
- Here's how it should be prototyped :

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

- It should return 1 if str is empty.

Chapter XII

Exercise 10 : ft_strlowercase

	Exercise 10
	ft_strlowercase
	Turn-in directory : ex10/
	Files to turn in : ft_strlowercase.c

- Create a function that transforms every uppercase letter of the string to lowercase.
- Here's how it should be prototyped :

```
char *ft_strlowercase(char *str);
```

- It should return str.

Chapter XIII

Exercise 11 : ft_strcpy

	Exercise 11
	ft_strcpy
	Turn-in directory : ex11/
	Files to turn in : ft_strcpy.c

- Reproduce the behavior of the function strcpy (man strcpy).
- Here's how it should be prototyped :

```
char *ft_strcpy(char *dest, char *src);
```

Chapter XIV

Exercise 12 : ft_ft

	Exercise 12
	ft_ft
Turn-in directory : ex12/	
Files to turn in : ft_ft.c	

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

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

Chapter XV

Exercise 13 : ft_swap

	Exercise 13
	ft_swap
	Turn-in directory : ex13/
	Files to turn in : ft_swap.c

- Create a function that swaps the values of two integers using pointers.
- Here's how it should be prototyped :

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

Chapter XVI

Exercise 14 : ft_foreach

	Exercise 14
	ft_foreach
	Turn-in directory : ex14/
	Files to turn in : ft_foreach.c

- Create a function that applies the function f to each element of the integer array.
- Here's how it should be prototyped :

```
void ft_foreach(int *tab, int length, void (*f)(int));
```

Chapter XVII

Exercise 15 : ft_print_param

	Exercise 15
	ft_print_param
	Turn-in directory : ex15/
	Files to turn in : ft_print_param.c

- Create a program that displays all its arguments, one per line.
- Here's how the main should be prototyped :

```
int main(int argc, char **argv);
```

Chapter XVIII

Exercise 16 : ft_main_printf

	Exercise 16
	ft_main_printf
	Turn-in directory : ex16/
	Files to turn in : ft_main_printf.c

- Using only the variables and functions already declared below, display the following output using printf. You may not use any other function or value.
- Here's how it should be called :

```
./a.out | cat -e
```

- The output should be :

```
Hello world ! c'est 42$
```

- Here is the provided code :

```
int mul2(int nb){return (nb * 2);}
char upper(char c){return (c - 32);}
char *add1(char str[]){int i = 0; while(str[i])
    {if (str[i] != 32){str[i] += 1;} i++;} return str;}
char *rm1(char str[]){int i = 0; while(str[i])
    {if (str[i] != 32){str[i] -= 1;} i++;} return str;}

int main(void)
{
    int nb = '*';
    char c = 99;
    char e = 33;
    char str[] = "Gdkkn vngkc";
    char str2[] = "(ftu";
    char d = 34;
    int nbr = 43;
    char str3[] = "Gdkqwe weqss !";
    char str4[] = "*fweeu";
}
```

Note Exceptionally for this exercise, the code is not required to follow the Norm, and does not need to compile with the -Wall -Wextra -Werror flags.