



C Piscine

C 03

Summary: This document is the subject for the C 03 module of the C Piscine @ 42.

Version: 3.2

Contents

I	Instructions	2
II	Foreword	4
III	Exercise 00 : ft_strcmp	5
IV	Exercise 01 : ft_strncmp	6
V	Exercise 02 : ft_strcat	7
VI	Exercise 03 : ft_strncat	8
VII	Exercise 04 : ft_strstr	9
VIII	Exercise 05 : ft_strlcat	10
IX	Submission and peer-evaluation	11

Chapter I

Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change 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 `norminette` to check if your files respect the norm. TL;DR: it would be idiotic to submit a piece of work that doesn't pass `norminette`'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.
- 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 `cc`.
- 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, or the slack Piscine.
- 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 !!!



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

Chapter II

Foreword

The first known mention of the game of RPS was in the book *Wuzazu* written by the Chinese Ming-dynasty writer Xie Zhaozhi who wrote that the game dated back to the time of the Chinese Han dynasty (206 BC – 220 AD). In the book, the game was called *shoushiling*. Li Rihua's book *Note of Liuyanzhai* also mentions this game, calling it *shoushiling*, *huozhitou*, or *huoquan*.

Throughout Japanese history there are frequent references to “*sansukumi-ken*”, meaning “ken” fist games with a “san” three-way “sukumi” deadlock. This is in the sense that A beats B, B beats C, and C beats A. The games originated in China before being imported to Japan and subsequently becoming popular.


By the early 20th century, rock–paper–scissors had spread beyond Asia, especially through increased Japanese contact with the west. Its English-language name is therefore taken from a translation of the names of the three Japanese hand-gestures for rock, paper and scissors: elsewhere in Asia the open-palm gesture represents “cloth” rather than “paper”. The shape of the scissors is also adopted from the Japanese style.

In 1927 *La Vie au patronage*, a children's magazine in France, described it in detail, referring to it as a “*jeu japonais*” (“Japanese game”). Its French name, “*Chi-fou-mi*”, is based on the Old Japanese words for “one, two, three” (“hi, fu, mi”)

A New York Times article of 1932 on the Tokyo rush hour describes the rules of the game for the benefit of American readers, suggesting it was not at that time widely known in the U.S. The 1933 edition of the *Compton's Pictured Encyclopedia* described it as a common method of settling disputes between children in its article on Japan; the name was given as “*John Kem Po*” and the article pointedly asserted, “This is such a good way of deciding an argument that American boys and girls might like to practice it too.”

Chapter III

Exercise 00 : ft_strcmp


	Exercise 00
ft_strcmp	
Turn-in directory : <i>ex00/</i>	
Files to turn in : ft_strcmp.c	
Allowed functions : None	

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

```
int      ft_strcmp(char *s1, char *s2);
```

Chapter IV

Exercise 01 : ft_strncmp


	Exercise 01
	ft_strncmp
Turn-in directory : <i>ex01/</i>	
Files to turn in : ft_strncmp.c	
Allowed functions : None	

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

```
int      ft_strncmp(char *s1, char *s2, unsigned int n);
```

Chapter V

Exercise 02 : ft_strcat

	Exercise 02
	ft_strcat
	Turn-in directory : ex02/
	Files to turn in : ft_strcat.c
	Allowed functions : None

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

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

- Appends string to destination
- Overwrites '\0' of dest
- When finished, adds an '\0' at the end of dest
- NULL?

↓ ↓ ↓ ↓
[ABC] [DEF]
↑ ↑ ↑

```
char *ft_strcat(char *dest, char *src)
{
    char *original;
    original = dest;


    while (*dest)
        dest++;

    while (*src) {
        *dest = *src;
        dest++;
        src++;
    }

    *dest = '\0';
    return (original);
}
```


Chapter VI

Exercise 03 : ft_strncat

	Exercise 03
	ft_strncat
Turn-in directory : <i>ex03/</i>	
Files to turn in : ft_strncat.c	
Allowed functions : None	

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


```
char *ft_strncat(char *dest, char *src, unsigned int nb);
```

```
char *ft_strncat(char *dest, char *src, unsigned int nb)
{
    char *original;
    original = dest;

    while (*dest)
        dest++;
    while (*src & nb) {
        *dest = *src;
        dest++;
        src++;
        nb--;
    }
    *dest = '\0';
    return (original);
}
```

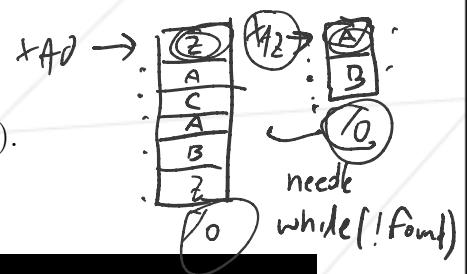
Chapter VII

Exercise 04 : ft_strstr

	Exercise 04
	ft_strstr
	Turn-in directory : ex04/
	Files to turn in : ft_strstr.c
	Allowed functions : None

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

```
char *ft_strstr(char *str, char *to_find);
```

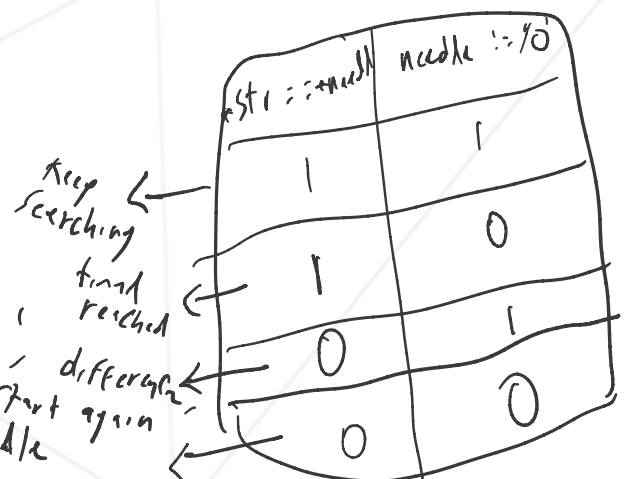


```
char *ft_strstr(char *str, char *to_find)
{
    while (*str && !found) {
```

```
        needle = to_find;
        while (*str == *needle && *needle != '\0') {
            *str++;
            *needle++;
```


```
        }
        if (*needle == '\0')
            found = 1;
```

```
        else
            needle = to_find;
        str++;
        needle;
```



Chapter VIII

Exercise 05 : ft_strlcat

	Exercise 05
ft_strlcat	
Turn-in directory : <i>ex05/</i>	
Files to turn in : ft_strlcat.c	
Allowed functions : None	

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

```
unsigned int ft_strlcat(char *dest, char *src, unsigned int size);
```

Chapter IX

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your files to ensure they are correct.



You need to return only the files requested by the subject of this project.