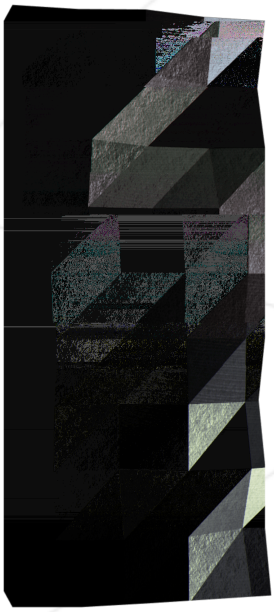

C Piscine

C 09

Summary: This document is the subject for the module C 09 of the C Piscine @ 42.

Contents

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 chea G i4s2ti1o7(elraded)-326(wi326(as)-327(on)-326(y)27feto)-326(folltia
-
-

ou6culti326(ase83)-ase83asi27(s)e83

8622(n5(doer)edures) T ET126.03412511.797 497.179 cm 0 d 0 0.393-1168 m 114.395 0 l S BT

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

Chapter II

Foreword

Dialog from the movie `The Big Lebowski`:

```
The Dude: Walter, ya know, it's Smokey, so his toe slipped over the line a little,
  big deal. It's just a game, man.
Walter Sobchak: Dude, this is a league game, this determines who enters the next
  round robin. Am I wrong? Am I wrong?
Smokey: Yeah, but I wasn't over. Gimme the marker Dude, I'm marking it 8.
Walter Sobchak: [pulls out a gun] Smokey, my friend, you are entering a world
  of pain.
The Dude: Walter...
Walter Sobchak: You mark that frame an 8, and you're entering a world of pain.
Smokey: I'm not...
Walter Sobchak: A world of pain.
Smokey: Dude, he's your partner...
Walter Sobchak: [shouting] Has the whole world gone crazy? Am I the only one
  around here who gives a shit about the rules? Mark it zero!
The Dude: They're calling the cops, put the piece away.
Walter Sobchak: Mark it zero!
[points gun in Smokey's face]
The Dude: Walter...
Walter Sobchak: [shouting] You think I'm fucking around here? Mark it zero!
Smokey: All right, it's fucking zero. Are you happy, you crazy fuck?
Walter Sobchak: ...It's a league game, Smokey.
```

Chapter III

Exercise 00 : libft

	Exercise 00
libft	
Turn-in directory : $ex00/$	
Files to turn in : `libft_creator.sh, ft_putchar.c, ft_swap.c, ft_putstr.c, ft_strlen.c, ft_strcmp.c`	
Allowed functions : `write`	

- Create your `ft` library. It'll be called `libft.a`.
- A shell script called `libft_creator.sh` will compile source files appropriately and will create your library.
- This library should contain all of the following functions :

```
void    ft_putchar(char c);
void    ft_swap(int *a, int *b);
void    ft_putstr(char *str);
int     ft_strlen(char *str);
int     ft_strcmp(char *s1, char *s2);
```

- We'll launch the following command-line :

```
sh libft_creator.sh
```

Chapter IV

Exercise 01 : Makefile

	Exercise 01
Makefile	
Turn-in directory : $ex01/$	
Files to turn in : `Makefile`	
Allowed functions : `None`	

- Create the `Makefile` that'll compile a library `libft.a`.
- Your makefile should print all the command it's running.
- Your makefile should not run any unnecessary command.
- The `Makefile` will get its source files from the "srcs" directory.
- Those files will be: ft_putchar.c, ft_swap.c, ft_putstr.c, ft_strlen.c, ft_strcmp.c
- The `Makefile` will get its header files from the "includes" directory.
- Those files will be: ft.h
- It should compile the .c files with gcc and with -Wall -Wextra -Werror flags in that order.
- The lib should be at the root of the exercise.
- .o files should be near their .c file.
- The `Makefile` should also implement the following rules: `clean`, `fclean`, `re`, `all` and of course `libft.a`.
- Running just `make` should be equal to `make all`
- The rule `all` should be equal to `make libft.a`.
- The rule `clean` should remove all the temporary generated files.

- The rule `fclean` should be like a `make clean` plus all the binary made with `make all`.
- The rule `re` should be like a `make fclean` followed by `make all`.
- Your makefile should not compile any file for nothing.
- We'll only fetch your Makefile and test it with our files.

`Watch out for wildcards!`

Chapter V

Exercise 02 : ft_split

	Exercise 02
ft_split	
Turn-in directory : $ex02/$	
Files to turn in : `ft_split.c`	
Allowed functions : `malloc`	

- Create a function that slits a string of character depending on another string of characters.
- You'll have to use each character from the string `charset` as a separator.
- The function returns an array where each box contains the address of a string wrapped between two separators. The last element of that array should equal to 0 to indicate the end of the array.
- There cannot be any empty strings in your array. Draw your conclusions accordingly.
- The string given as argument won't be modifiable.
- Here's how it should be prototyped :

```
char **ft_split(char *str, char *charset);
```