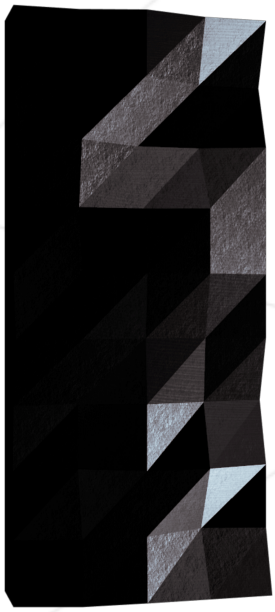

C Piscine

C 02

Summary: このドキュメントは*C Piscine @ 42*の *C 02*モジュール用の課題です。

Contents

I	Instructions	2
II	Foreword	4
III	Exercise 00 : ft_strcpy	5
IV	Exercise 01 : ft_strncpy	6
V	Exercise 02 : ft_str_is_alpha	7
VI	Exercise 03 : ft_str_is_numeric	8
VII	Exercise 04 : ft_str_is_lowercase	9
VIII	Exercise 05 : ft_str_is_uppercase	10
IX	Exercise 06 : ft_str_is_printable	11
X	Exercise 07 : ft_strupcase	12
XI	Exercise 08 : ft_strlowcase	13
XII	Exercise 09 : ft_strcapitalize	14
XIII	Exercise 10 : ft_strlcpy	15
XIV	Exercise 11 : ft_putstr_non_printable	16
XV	Exercise 12 : ft_print_memory	17

Chapter I

Instructions

- このページのみを参考にしてください。噂を信用しないで下さい。
- この書類は、提出前に変更になる可能性があります。十分に注意して下さい。
- ファイルとディレクトリへの権限があることをあらかじめ確認して下さい。
- 課題は全て提出手順に従って行って下さい。
- 課題の確認と評価は、あなたのクラスメイトが行います。
- 課題はMoulinetteと呼ばれるプログラムによっても確認・評価されます。
- Moulinetteは大変細かい評価を行います。全て自動で行われ、交渉方法はありません。頑張ってください。
- Moulinetteは規範を無視したコードは解読ができません。 Moulinetteはあなたのファイルが規範を遵守しているかをチェックするために、`norminette`と呼ばれるプログラムを使って判断します。要約：せっかくの取り組みが`norminette`のチェックによって無駄になるのは勿体無いので、気をつけましょう。
- 課題は簡単なものから徐々に難しくなるように並べられています。簡単な課題が解けていない場合、難しい問題かが解けていたとしても **加点されることはありません**。
- 禁止されている関数をしようした場合は不正とみなします。不正者は`-42`の評価をつけられこの評価に交渉の余地はありません。
- プログラムを要求する際はmain()関数のみを提出しましょう。
- Moulinetteはこれらのフラッグを用いて`gcc`でコンパイルします：-Wall -Wextra -Werror。
- プログラムか〝コンパイルされなかった場合、評価は`0`です。
- 課題で指定されているもの以外はどんなファイルもディレクトリ内に残しておくことはできません。
- 質問があれば右側の人に聞きましょう。それでも分からなければ左側の人に聞いてください。

- あなたを助けてくれるのはGoogle / 人間 / インターネット / ...と呼ばれているものです。
- intranet上のフォーラムの”C Piscine”パートかPiscineのslackを確認してください。
- 例を徹底的に調べてください。課題で言及されていない詳細まで要求されます。

Norminetteは、 *-R CheckForbiddenSourceHeader* をオプションに追加しなければなりません。その際、Moulinetteも使用します。

Chapter II

Foreword

Here is a discuss extract from the `Silicon Valley` serie:

```
- I mean, why not just use Vim over Emacs? (CHUCKLES)
- I do use Vim over Emac.
- Oh, God, help us! Okay, uh you know what? I just don't think this is
  going to work. I'm so sorry. Uh, I mean like, what, we're going to bring
  kids into this world with that over their heads? That's not really fair
  to them, don't you think?
- Kids? We haven't even slept together.
- And guess what, it's never going to happen now, because there is no
  way I'm going to be with someone who uses spaces over tabs.
- Richard! (PRESS SPACE BAR MANY TIMES)
- Wow. Okay. Goodbye.
- One tab saves you eight spaces! - (DOOR SLAMS) - (BANGING)

. . .

(RICHARD MOANS)

- Oh, my God! Richard, what happened?
- I just tried to go down the stairs eight steps at a time. I'm okay, though.
- See you around, Richard.
- Just making a point.
```

Hopefully, you are not forced to use emacs and your space bar to complete the following exercices.

Chapter III

Exercise 00 : ft_strcpy

	Exercise 00
ft_strcpy	
提出するディレクトリ：$ex00/$	
提出するファイル：`ft_strcpy.c`	
使用可能な関数：`None`	

- Reproduce the behavior of the function `strcpy` (man strcpy).
- プロトタイプ例

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

Chapter IV

Exercise 01 : ft_strncpy

	Exercise 01
ft_strncpy	
提出するディレクトリ：$ex01/$	
提出するファイル：`ft_strncpy.c`	
使用可能な関数：`None`	

- Reproduce the behavior of the function `strncpy` (man strncpy).
- プロトタイプ例

```
char	*ft_strncpy(char *dest, char *src, unsigned int n);
```

Chapter V

Exercise 02 : ft_str_is_alpha

	Exercise 02
ft_str_is_alpha	
提出するディレクトリ：$ex02/$	
提出するファイル：`ft_str_is_alpha.c`	
使用可能な関数：`None`	

- パラメータとして与えられた文字列が英字だけを含んでいる時は1を、他の文字が含まれている時は0を返す関数を作成しましょう。
- プロトタイプ例

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

- `str`が空の場合は1を返します。

Chapter VI

Exercise 03 : ft_str_is_numeric

	Exercise 03
ft_str_is_numeric	
提出するディレクトリ：$ex03/$	
提出するファイル：`ft_str_is_numeric.c`	
使用可能な関数：`None`	

- パラメータとして与えられた文字列が数字のみを含む場合は1を、それ以外の文字が含まれる場合には0を返す関数を作成しましょう。
- プロトタイプ例

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

- `str`が空の場合は1を返します。

Chapter VII

Exercise 04 : ft_str_is_lowercase

	Exercise 04
ft_str_is_lowercase	
提出するディレクトリ：$ex04/$	
提出するファイル：`ft_str_is_lowercase.c`	
使用可能な関数：`None`	

- パラメータとして与えられた文字列が小文字のアルファベットだけを含む場合は1を、その他の文字を含む場合は0を返す関数を作成しましょう。
- プロトタイプ例

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

- `str`が空の場合は1を返します。

Chapter VIII

Exercise 05 : ft_str_is_uppercase

	Exercise 05
ft_str_is_uppercase	
提出するディレクトリ：$ex05/$	
提出するファイル：`ft_str_is_uppercase.c`	
使用可能な関数：`None`	

- パラメータとして与えられた文字列が大文字のアルファベットだけを含む場合は1を、その他の文字を含む場合は0を返す関数を作成しましょう。
- プロトタイプ例

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

- `str`が空の場合は1を返します。

Chapter IX

Exercise 06 : ft_str_is_printable

	Exercise 06
ft_str_is_printable	
提出するディレクトリ：$ex06/$	
提出するファイル：`ft_str_is_printable.c`	
使用可能な関数：`None`	

- パラメータとして与えられた文字列が表示文字だけを含む場合は1を、その他の文字を含む場合は0を返す関数を作成しましょう。
- プロトタイプ例

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

- `str`が空の場合は1を返します。

Chapter X

Exercise 07 : ft_strupcase

	Exercise 07
ft_strupcase	
提出するディレクトリ：$ex07/$	
提出するファイル：`ft_strupcase.c`	
使用可能な関数：`None`	

- 文字列にある全ての文字を大文字に変換する関数を作成しましょう。
- プロトタイプ例

```
char        *ft_strupcase(char *str);
```

- `str`を返します。

Chapter XI

Exercise 08 : ft_strlowcase

	Exercise 08
ft_strlowcase	
提出するディレクトリ：$ex08/$	
提出するファイル：`ft_strlowcase.c`	
使用可能な関数：`None`	

- 文字列にある全ての文字を小文字に変換する関数を作成しましょう。
- プロトタイプ例

```
char        *ft_strlowcase(char *str);
```

- `str`を返します。

Chapter XII

Exercise 09 : ft_strcapitalize

	Exercise 09
ft_strcapitalize	
提出するディレクトリ：$ex09/$	
提出するファイル：`ft_strcapitalize.c`	
使用可能な関数：`None`	

- 各単語の1文字目を大文字にし、それ以降の文字列を全て小文字に変換する関数を作成しましょう。
- 単語は英数字の文字列です。
- プロトタイプ例

```
char        *ft_strcapitalize(char *str);
```

- `str`を返します。
- 例

```
salut, comment tu vas ? 42mots quarante-deux; cinquante+et+un
```

- は、このようになります。

```
Salut, Comment Tu Vas ? 42mots Quarante-Deux; Cinquante+Et+Un
```

Chapter XIII

Exercise 10 : ft_strlcpy

	Exercise 10
ft_strlcpy	
提出するディレクトリ：$ex10/$	
提出するファイル：`ft_strlcpy.c`	
使用可能な関数：`None`	

- Reproduce the behavior of the function `strlcpy` (man strlcpy).
- プロトタイプ例

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

Chapter XIV

Exercise 11 : ft_putstr_non_printable

	Exercise 11
ft_putstr_with_non_printable	
提出するディレクトリ：$ex11/$	
提出するファイル：`ft_putstr_non_printable.c`	
使用可能な関数：`write`	

- 画面に文字列を表示する関数を作成しましょう。文字列に表示不可能な文字がある場合は16進数（小文字）にして、その文字の前に "バックスラッシュ"をつけて表示します。
- 例

```
Coucou\ntu vas bien ?
```

- 表示例

```
Coucou\0atu vas bien ?
```

- プロトタイプ例

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

Chapter XV

Exercise 12 : ft_print_memory

	Exercise 12
ft_print_memory	
提出するディレクトリ：$ex12/$	
提出するファイル：`ft_print_memory.c`	
使用可能な関数：`write`	

- 画面にメモリ領域を表示する関数を作成しましょう。
- メモリ領域の表示は3つの列に分かれていて、スペースで区切られています。
  - 1行目の1文字目の16進数アドレスと':'
  - 2文字ごとにスペースを含む16進数は必要に応じてスペースを入れましょう。(下記の例参照)
  - アドレスにある要素を表示文字に変換
- 文字が表示文字ではない場合、ドット(.')に置き換えます。
- 各行は16文字、処理します。
- サイズが0の場合は何も表示しないでください。

- 例

```
$> ./ft_print_memory
000000010a161f40: 426f 6e6a 6f75 7220 6c65 7320 616d 696e Bonjour les amin
000000010a161f50: 6368 6573 090a 0963 2020 6573 7420 666f ches...c est fo
000000010a161f60: 7509 746f 7574 0963 6520 7175 206f 6e20 u.tout.ce qu on
000000010a161f70: 7065 7574 2066 6169 7265 2061 7665 6309 peut faire avec.
000000010a161f80: 0a09 7072 696e 745f 6d65 6d6f 7279 0a0a ..print_memory..
000000010a161f90: 0a09 6c6f 6c2e 6c6f 6c0a 2000      ..lol.lol. .
$> ./ft_print_memory | cat -te
0000000107ff9f40: 426f 6e6a 6f75 7220 6c65 7320 616d 696e Bonjour les amin$
0000000107ff9f50: 6368 6573 090a 0963 2020 6573 7420 666f ches...c est fo$
0000000107ff9f60: 7509 746f 7574 0963 6520 7175 206f 6e20 u.tout.ce qu on $
0000000107ff9f70: 7065 7574 2066 6169 7265 2061 7665 6309 peut faire avec.$
0000000107ff9f80: 0a09 7072 696e 745f 6d65 6d6f 7279 0a0a ..print_memory..$
0000000107ff9f90: 0a09 6c6f 6c2e 6c6f 6c0a 2000      ..lol.lol. .$
$>
```

- プロトタイプ例

```
void        *ft_print_memory(void *addr, unsigned int size);
```

- addr を返します。