



C Piscine

C 04

Summary: このドキュメントは、42の *C Piscine C 04* モジュール用の課題です。

Version: 5

Contents

I	Instructions	2
II	AI Instructions	4
III	Foreword	7
IV	Exercise 00 : ft_strlen	9
V	Exercise 01 : ft_putstr	10
VI	Exercise 02 : ft_putnbr	11
VII	Exercise 03 : ft_atoi	12
VIII	Exercise 04 : ft_putnbr_base	13
IX	Exercise 05 : ft_atoi_base	15
X	Submission and peer-evaluation	16

Chapter I

Instructions

- このページのみが正式な課題の指示です。噂に惑わされないようにしてください。
- この課題は提出前に変更される可能性がありますので、気をつけてください！
- ファイルとディレクトリへの適切な権限があることを確認してください。
- すべての課題で、提出手順に従ってください。
- 提出した課題の評価（レビュー）は、あなたの周りにはいるPiscine受験者によって行われます。
- さらに、**Moulinette**（自動採点プログラム）にも評価されます。
- **Moulinette**は非常に厳格かつ厳密に評価を行います。完全に自動化されており、交渉の余地はありません。思わぬ減点を避けるためにも、細部まで丁寧に取り組んでください。
- **Moulinette**は融通が利きません。あなたのコードがNormに準拠していない場合、理解しようとしません。**Moulinette**は**norminette**というプログラムを使用して、ファイルがNormに準拠しているかチェックします。要約：**norminette**のチェックに通らない課題を提出しても意味がありません。
- 課題は難易度が低いものから高いものへと難易度順に並んでいます。前の課題が正しく解けていない場合、後の課題が解けていても評価されません。
- 禁止されている関数を使用することは不正行為とみなされます。不正行為者は**-42**という成績がつけられ、これは交渉の余地がありません。
- 私たちが具体的にプログラムを求める場合のみ、**main()**関数を提出する必要があります。
- **Moulinette**は以下のフラグでコンパイルします：**-Wall -Wextra -Werror, cc**を使用します。
- プログラムがコンパイルできない場合、**0**の評価になります。
- 課題で指定されたファイル以外は、ディレクトリに絶対に残さないでください。

- 質問がある場合は、まず右隣の仲間に聞いてみましょう。いなければ左隣の仲間に聞いてみてください。
- あなたの助けになる参考資料はGoogle / man / インターネット / ...です。
- Intranetのフォーラムの「C Piscine」セクションをチェックしてください。
- 課題の例（出力例）をよく読んでください。課題の説明文だけではすぐに分からない要件が隠れていることがあります。
- 頭を使ってください!!!



Norminetteは、`-R CheckForbiddenSourceHeader` をオプションに追加しなければなりません。Moulinetteも同様にこのオプションを使用します。

Chapter II

AI Instructions

● Context

C Piscineは強烈な体験です。42で迎える最初の大きな挑戦として、問題解決、自律性、そしてコミュニティの世界に深く飛び込むことになります。

このフェーズの主な目的は、試行錯誤や反復、そして特にピアラーニングを通じた交流によって、自分の土台を築き上げることです。

AI時代において、近道を見つけることは簡単です。しかし、AIの利用が本当にあなたの成長を助けているのか、それとも真のスキルを身につける上で障害となっていないか、よく考えることが大切です。

Piscineはまたとない人との繋がりを経験する場でもあります。今のところ、それにとって代わるものは何ともありません。AIでさえも。

学習ツールとして、ICTカリキュラムの一環として、そして労働市場での期待の高まりに応えるものとして、私たちのAIに対する考え方の詳細については、Intranet上の専用FAQをご覧ください。

● 主なメッセージ

- 近道をせず、強固な基礎を築く。
- 技術力と実践力を真に養う。
- 真のピアラーニングを体験し、学び方や新たな問題の解決方法を学び始める。
- 結果よりも学習の過程が重要である。
- AIに伴うリスクを学び、一般的な落とし穴を避けるための効果的な管理方法と対策を身につける。

● 学習者のルール:

- 課題には、AIに頼る前にまず自分の頭で考えること。
- AIに直接的な答えを求めないこと。
- AIに対する42のグローバルな方針を学ぶこと。

● このフェーズでの到達目標:

この基礎フェーズを終えることで、以下の点を達成できます。

- 技術とコーディングの適切な基礎を身につける。
- この段階においてAIが危険となりうる理由と、その危険性を理解する。

● コメントと具体例:

- 私たちはAIの存在を知っていますし、それがあなたのプロジェクトを解決できることも知っています。しかし、あなたは学ぶためにここにいるのであって、AIが学習したことを証明するためではありません。AIが特定の問題を解決できることを示すためだけに、あなた（や私たち）の時間を無駄にしないでください。
- 42での学習は、答えを知ることではありません。答えを見つけ出す能力を養うことです。AIは直接答えを与えてくれますが、それはあなた自身の論理的思考を妨げます。そして、論理的思考には時間と努力、そして失敗が伴います。成功への道は、決して楽なものではありません。
- Exam中はAIが利用できないことを心に留めておいてください。インターネットもスマートフォンもありません。学習過程でAIに過度に頼りすぎている場合、すぐにそのことに気づくでしょう。
- ピアラーニングは、多様な考え方やアプローチに触れる機会を与え、対人スキルや多角的な思考能力を向上させます。これは、単にボットとチャットするよりもはるかに価値があります。ですから、恥ずかしがらずに、話しかけ、質問し、共に学びましょう！
- AIは学習ツールとして、またそれ自体がトピックとして、カリキュラムの一部になります。独自のAIソフトウェアを開発する機会さえあります。私たちの段階的なアプローチについてさらに学ぶためには、Intranetで利用可能なドキュメントを参照してください。

✓ 良い実践例:

新しいコンセプトの学習でつまづいています。近くの人にどうアプローチしたか尋ねてみます。10分ほど話していると、突然ひらめき、理解できました。

✕ 悪い実践例:

こっそりAIを使い、良さそうなコードをコピーします。ピアレビューの際には何も説明できず、不合格になりました。ExamではAIがなく、また行き詰まり、不合格になりました。

Chapter III

Foreword

Here are the lyrics for City Hunter' s theme song "Moonlight Shadow":

The last time ever she saw him
Carried away by a moonlight shadow
He passed on worried and warning
Carried away by a moonlight shadow.
Lost in a riddle that Saturday night
Far away on the other side.
He was caught in the middle of a desperate fight
And she couldn't find how to push through

The trees that whisper in the evening
Carried away by a moonlight shadow
Sing a song of sorrow and grieving
Carried away by a moonlight shadow
All she saw was a silhouette of a gun
Far away on the other side.
He was shot six times by a man on the run
And she couldn't find how to push through

[Chorus]
I stay, I pray
See you in Heaven far away...
I stay, I pray
See you in Heaven one day.

Four A.M. in the morning
Carried away by a moonlight shadow
I watched your vision forming
Carried away by a moonlight shadow
A star was glowing in the silvery night
Far away on the other side
Will you come to talk to me this night
But she couldn't find how to push through


[Chorus]

Far away on the other side.
Caught in the middle of a hundred and five
The night was heavy and the air was alive
But she couldn't find how to push through
Carried away by a moonlight shadow
Carried away by a moonlight shadow
Far away on the other side.

Unfortunately, this module has nothing to do with City Hunter.

Chapter IV

Exercise 00 : ft_strlen


	Exercise 00
	ft_strlen
	提出するディレクトリ : <i>ex00/</i>
	提出するファイル : ft_strlen.c
	使用可能な関数 : None

- 文字列にある文字数を数えて返す関数を作成してください。
- プロトタイプ例 :

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

Chapter V

Exercise 01 : ft_putstr


	Exercise 01
	ft_putstr
	提出するディレクトリ : <i>ex01/</i>
	提出するファイル : <i>ft_putstr.c</i>
	使用可能な関数 : <i>write</i>

- 標準出力に文字列を出力する関数を作成してください。
- プロトタイプ例 :

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

Chapter VI

Exercise 02 : ft_putnbr

	Exercise 02
	ft_putnbr
	提出するディレクトリ : <i>ex02/</i>
	提出するファイル : <i>ft_putnbr.c</i>
	使用可能な関数 : <i>write</i>


- 引数として渡された数字を標準出力に出力する関数を作成してください。その際、`int`型に収まるすべてのパターンを出力してください。
- プロトタイプ例 :

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

- 例 :
 - `ft_putnbr(42)` は "42" を出力します。

Chapter VII

Exercise 03 : ft_atoi

	Exercise 03
	ft_atoi
	提出するディレクトリ : <i>ex03/</i>
	提出するファイル : <i>ft_atoi.c</i>
	使用可能な関数 : None

- `str`が指定する文字列の最初の部分を整数に変換する関数を作成してください。
- 文字列は任意の数の空白文字で始まる場合があります (`isspace(3)`で定義される空白文字)。
- 続いて、任意の数の '+' や '-' 記号が現れる場合があります。 '-' 記号の数が奇数か偶数かによって、結果の符号が決まります。
- その後、関数は連続する数字を10進数で処理するようにしてください。
- 関数は、条件に合わない文字が見つかるまで文字列を読み込み、そこまでに変換した数値を返します。
- オーバーフローとアンダーフローは処理しないでください。それらが発生した場合、関数の戻り値は未定義となります。
- 実行例 :


```
$>./a.out " ---+---+1234ab567"  
-1234
```

- プロトタイプ例 :

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

Chapter VIII

Exercise 04 : ft_putnbr_base

	Exercise 04
	ft_putnbr_base
	提出するディレクトリ : <i>ex04/</i>
	提出するファイル : <i>ft_putnbr_base.c</i>
	使用可能な関数 : <i>write</i>


- 整数をベースシステムで標準出力に出力する関数を作成してください。
- 数字は`int`で与えられ、ベースは文字列で与えられます。
- ベースシステムは、数字を出力するために使用できるすべての記号から構成されています。
 - 0123456789は、10進数を出力するために一般的に使用されるベースシステムです。
 - 01は、バイナリーベースシステムです。
 - 0123456789ABCDEFは、16進数のベースシステムです。
 - poneyvifは、8進数のベースシステムです。
- 関数は負の数にも対応する必要があります。
- 無効な引数が与えられた場合は、標準出力に何も出力しないでください。
無効な引数の例：
 - ベースが空、または、1のサイズである場合
 - ベースに同じ文字が2つ含まれている場合
 - ベースに '+' または '-' が含まれている場合

- プロトタイプ例：

```
void      ft_putnbr_base(int nbr, char *base);
```

Chapter IX

Exercise 05 : ft_atoi_base

	Exercise 05
	ft_atoi_base
	提出するディレクトリ : <i>ex05/</i>
	提出するファイル : <i>ft_atoi_base.c</i>
	使用可能な関数 : None

- `str`が指定する文字列の最初を整数に変換する関数を作成してください。
- `str`は、2つ目の引数として与えられた特定のベースで表されている文字列です。
- ベースに関するルールを除いて、関数は`ft_atoi`と同じように動作する必要があります。
- 無効な引数が与えられた場合は、0を返すようにしてください。
無効な引数の例：
 - ベースが空、または、1のサイズである場合
 - ベースに同じ文字が2つ含まれている場合
 - ベースに‘+’または‘-’、空白文字が含まれている場合
- プロトタイプ例：

```
int ft_atoi_base(char *str, char *base);
```


Chapter X

Submission and peer-evaluation

課題は、いつも通りGitリポジトリに提出してください。リポジトリ内の提出物のみが、レビュー中の評価対象となります。ファイル名が正しいかどうか、必ず確認してください。



この課題の指示で、明示的に求められたファイルのみを提出してください。