



C Piscine

C 012

*Summary:* このドキュメントはC Piscine @ 42の C 12モジュール用の課題です。

# Contents

I	Foreword	2
II	Instructions	4
III	Exercice 00 : ft_create_elem	6
IV	Exercice 01 : ft_list_push_front	7
V	Exercice 02 : ft_list_size	8
VI	Exercice 03 : ft_list_last	9
VII	Exercice 04 : ft_list_push_back	10
VIII	Exercice 05 : ft_list_push_strs	11
IX	Exercice 06 : ft_list_clear	12
X	Exercice 07 : ft_list_at	13
XI	Exercice 08 : ft_list_reverse	14
XII	Exercice 09 : ft_list_foreach	15
XIII	Exercice 10 : ft_list_foreach_if	16
XIV	Exercice 11 : ft_list_find	17
XV	Exercice 12 : ft_list_remove_if	18
XVI	Exercice 13 : ft_list_merge	19
XVII	Exercice 14 : ft_list_sort	20
XVIII	Exercice 15 : ft_list_reverse_fun	21
XIX	Exercice 16 : ft_sorted_list_insert	22
XX	Exercice 17 : ft_sorted_list_merge	23

# **Chapter I**

## **Foreword**

SPOILER ALERT  
DON'T READ THE NEXT PAGE

## You've been warned.

- In Star Wars, Dark Vador is Luke's Father.
- In The Usual Suspects, Verbal is Keyser Soze.
- In Fight Club, Tyler Durden and the narrator are the same person.
- In Sixth Sens, Bruce Willis is dead since the beginning.
- In The others, the inhabitants of the house are ghosts and vice-versa.
- In Bambi, Bambi's mother dies.
- In The Village, monsters are the villagers and the movie actually takes place in our time.
- In Harry Potter, Dumbledore dies.
- In Planet of apes, the movie takes place on earth.
- In Game of thrones, Robb Stark and Joffrey Baratheon die on their wedding day.
- In Twilight, Vampires shine under the sun.
- In Stargate SG-1, Season 1, Episode 18, O'Neill and Carter are in Antartica.
- In The Dark Knight Rises, Miranda Tate is Talia Al'Gul.
- In Super Mario Bros, The princess is in another castle.

# Chapter II

## 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：せっきくの取り組みが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。
- プログラムが `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。
- 今後の問題では以下の構造体を使用してください。

```
typedef struct          s_list
{
    struct s_list      *next;
    void                *data;
}                      t_list;
```

- この構造体をft\_list.hファイル内に含めて、各課題を提出しましょう。
- exercise 01より先はスタッフのft\_create\_elemを使用しますので、あらかじめ準備しておいてください。(プロトタイプをft\_list.hファイルに入れておくと便利です)。

# Chapter III

## Exercice 00 : ft\_create\_elem


	Exercise 00
	ft_create_elem
	提出するディレクトリ : <i>ex00/</i>
	提出するファイル : <i>ft_create_elem.c</i> , <i>ft_list.h</i>
	使用可能な関数 : <i>malloc</i>

- *t\_list*型の新しい要素を作成する、*ft\_create\_elem*関数を作成しましょう。
- *data*に与えられた引数を受け渡し、*next*にNULLを割り当てます。
- プロトタイプ例

```
t_list *ft_create_elem(void *data);
```

# Chapter IV

## Exercice 01 : ft\_list\_push\_front

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


- `t_list`型の新しい要素をリストのはじめに追加する`ft_list_push_front`関数を作成しましょう。
- `data`に与えられた引数を割り当てましょう。
- 必要があれば、リストのはじめのポインタを更新しましょう。
- プロトタイプ例

```
void ft_list_push_front(t_list **begin_list, void *data);
```



# Chapter V

## Exercice 02 : ft\_list\_size


	Exercise 02
	ft_list_size
	提出するディレクトリ : <i>ex02/</i>
	提出するファイル : <i>ft_list_size.c, ft_list.h</i>
	使用可能な関数 : None

- リスト内の要素数を返す、`ft_list_size`という関数を作成しましょう。
- プロトタイプ例

```
int ft_list_size(t_list *begin_list);
```

# Chapter VI

## Exercice 03 : ft\_list\_last


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

- リストの最後の要素を返す、`ft_list_last`という関数を作成しましょう。
- プロトタイプ例

```
t_list *ft_list_last(t_list *begin_list);
```

# Chapter VII

## Exercice 04 : ft\_list\_push\_back


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

- `t_list`型の新しい要素をリストの最後に追加する`ft_list_push_back`を作成しましょう。
- `data`に与えられた引数を割り当てましょう。
- 必要があれば、リストのはじめのポインタを更新しましょう。
- プロトタイプ例

```
void      ft_list_push_back(t_list **begin_list, void *data);
```

# Chapter VIII

## Exercice 05 : ft\_list\_push\_strs


	Exercise 05
	ft_list_push_strs
	提出するディレクトリ : <i>ex05/</i>
	提出するファイル : <i>ft_list_push_strs.c, ft_list.h</i>
	使用可能な関数 : <i>ft_create_elem</i>

- `strs`が指す全ての文字列を含む新しいリストを作成する`ft_list_push_strs`関数を作成しましょう。
- `size`は`strs`配列の要素数です。
- はじめの要素はリストの最後になるはずです。
- リスト内の最初のリンクのアドレスが返されます。
- プロトタイプ例

```
t_list *ft_list_push_strs(int size, char **strs);
```

# Chapter IX

## Exercice 06 : ft\_list\_clear


	Exercise 06
	ft_list_clear
	提出するディレクトリ : ex06/
	提出するファイル : ft_list_clear.c, ft_list.h
	使用可能な関数 : free

- リストから全てのリンクを削除し動的に確保されたメモリを解放する、ft\_list\_clear という関数を作成しましょう。
- free\_fctはdata用に動的に確保されたメモリを解放します。
- プロトタイプ例

```
void ft_list_clear(t_list *begin_list, void (*free_fct)(void *));
```

# Chapter X

## Exercice 07 : ft\_list\_at


	Exercise 07
	ft_list_at
	提出するディレクトリ : <i>ex07/</i>
	提出するファイル : <i>ft_list_at.c, ft_list.h</i>
	使用可能な関数 : None

- リストのN番目の要素を返す、ft\_list\_at関数を作成しましょう。リストの最初の要素は0番目です。
- エラーの際は、nullポインタを返します。
- プロトタイプ例

```
t_list *ft_list_at(t_list *begin_list, unsigned int nbr);
```

# Chapter XI

## Exercice 08 : ft\_list\_reverse


	Exercise 08
	ft_list_reverse
	提出するディレクトリ : <i>ex08/</i>
	提出するファイル : <i>ft_list_reverse.c</i>
	使用可能な関数 : None

- リスト内の要素の順番を反対にする、`ft_list_reverse`関数を作成しましょう。その際、各要素の値は同じままにしましょう。
- この関数をテストする際は我々の`ft_list.h`を使うので注意してください。
- プロトタイプ例

```
void ft_list_reverse(t_list **begin_list);
```

# Chapter XII

## Exercice 09 : ft\_list\_foreach

	Exercise 09
	ft_list_foreach
	提出するディレクトリ : <i>ex09/</i>
	提出するファイル : <i>ft_list_foreach.c, ft_list.h</i>
	使用可能な関数 : None

- リストの各リンクの要素に引数として与えられた関数を適用する、ft\_list\_foreach関数を作成しましょう。
- fはリストと同じ順番で適用されます。
- プロトタイプ例

```
void ft_list_foreach(t_list *begin_list, void (*f)(void *));
```


- f関数ポインタは次のように使用されます。

```
(*f)(list_ptr->data);
```



# Chapter XIII

## Exercise 10 : ft\_list\_foreach\_if

	Exercise 10
	ft_list_foreach_if
	提出するディレクトリ : ex10/
	提出するファイル : ft_list_foreach_if.c, ft_list.h
	使用可能な関数 : None

- 引数として渡された関数をリストのいくつかの要素に適用する、ft\_list\_foreach\_if関数を作成しましょう。
- cmpがdata\_refと要素を比較し0を返し場合のみ、f関数を適用しなさい。
- fはリストと同じ順番で適用されます。
- プロトタイプ例

```
void ft_list_foreach_if(t_list *begin_list, void (*f)(void *), void  
*data_ref, int (*cmp)())
```

- fとcmp関数ポインタは次のように使用されます。


```
(*f)(list_ptr->data);  
(*cmp)(list_ptr->data, data_ref);
```



例えば、cmp関数ポインタはft\_strcmp関数に成り得ます...

# Chapter XIV

## Exercice 11 : ft\_list\_find

	Exercise 11
	ft_list_find
	提出するディレクトリ : ex11/
	提出するファイル : ft_list_find.c, ft_list.h
	使用可能な関数 : None

- cmpとdata\_refを比較してデータを持つリストの全ての要素を返し、 cmp が 0を返すようにする関数、 ft\_list\_findを作成しましょう。
- プロトタイプ例


```
t_list *ft_list_find(t_list *begin_list, void *data_ref, int (*cmp)());
```

- cmp関数ポインタは次のように使用されます。

```
(*cmp)(list_ptr->data, data_ref);
```

# Chapter XV

## Exercice 12 : ft\_list\_remove\_if

	Exercise 12
	ft_list_remove_if
	提出するディレクトリ : ex12/
	提出するファイル : ft_list_remove_if.c, ft_list.h
	使用可能な関数 : free

- cmp関数がdata\_refと要素のdataを比較して0を返した場合、その要素をリストから削除するft\_list\_remove\_if関数を作成しましょう。
- free\_fctを使って要素のdataのメモリを解放してください。
- プロトタイプ例


```
void ft_list_remove_if(t_list **begin_list, void *data_ref, int (*cmp)(), void (*free_fct)(void *))
```

- cmpとfree\_fct関数ポインタは次のように使用されます。

```
(*cmp)(list_ptr->data, data_ref);  
(*free_fct)(list_ptr->data);
```

# Chapter XVI

## Exercice 13 : ft\_list\_merge


	Exercise 13
	ft_list_merge
	提出するディレクトリ : <i>ex13/</i>
	提出するファイル : <i>ft_list_merge.c, ft_list.h</i>
	使用可能な関数 : None

- begin2リストの要素を、begin1リストの末尾に置くft\_list\_merge関数を作成しましょう。
- 要素の作成は許可されていません。
- プロトタイプ例

```
void ft_list_merge(t_list **begin_list1, t_list *begin_list2);
```

# Chapter XVII

## Exercice 14 : ft\_list\_sort

	Exercise 14
	ft_list_sort
	提出するディレクトリ : ex14/
	提出するファイル : ft_list_sort.c, ft_list.h
	使用可能な関数 : None

- 2つの要素が持っているdataを比較できる関数を用いて、リスト内の要素を昇順に並び替えるft\_list\_sort関数を作成しましょう。
- プロトタイプ例

```
void ft_list_sort(t_list **begin_list, int (*cmp)());
```

- cmp関数ポインタは次のように使用されます。


```
(*cmp)(list_ptr->data, list_other_ptr->data);
```



例えば、cmp関数ポインタはft\_strcmp関数に成り得ます...

# Chapter XVIII

## Exercice 15 : ft\_list\_reverse\_fun


	Exercise 15
	ft_list_reverse_fun
	提出するディレクトリ : <i>ex15/</i>
	提出するファイル : <i>ft_list_reverse_fun.c, ft_list.h</i>
	使用可能な関数 : None

- リスト内の要素の順番を反対にするft\_list\_reverse\_fun関数を作成しましょう。
- プロトタイプ例

```
void ft_list_reverse_fun(t_list *begin_list);
```

# Chapter XIX

## Exercice 16 : ft\_sorted\_list\_insert

	Exercise 16
	ft_sorted_list_insert
	提出するディレクトリ : ex16/
	提出するファイル : ft_sorted_list_insert.c, ft_list.h
	使用可能な関数 : ft_create_elem

- 新しい要素を作成し、それを昇順をキープしたままリストに挿入する関数、`ft_sorted_list_insert` を作成しましょう。その際の並びは昇順のままです。
- プロトタイプ例


```
void ft_sorted_list_insert(t_list **begin_list, void *data, int (*cmp)());
```

- `cmp`関数ポインタは次のように使用されます。

```
(*cmp)(list_ptr->data, list_other_ptr->data);
```

# Chapter XX

## Exercice 17 : ft\_sorted\_list\_merge

	Exercise 17
	ft_sorted_list_merge
	提出するディレクトリ : ex17/
	提出するファイル : ft_sorted_list_merge.c, ft_list.h
	使用可能な関数 : None

- 並び替え済みのリストbegin2を並び替えず済みのリストであるbegin1と統合する関数、ft\_sorted\_list\_mergeを作成しましょう、その際、begin1は昇順で並び替えられたままの状態です。
- プロトタイプ例

```
void ft_sorted_list_merge(t_list **begin_list1, t_list *begin_list2, int (*cmp)());
```

- cmp関数ポインタは次のように使用されます。

```
(*cmp)(list_ptr->data, list_other_ptr->data);
```