

## С Пискин

День 11

Sta далее 42 pedago@42.fr

Аннотация: Данный документ является предметом Day11 на C Piscine @ 42.

#### содержание

Я	инструкции	2
II	предисловие	4
Ш	Exercice 00: ft_create_elem	6
IV Exe	rcice 01: ft_list_push_back	7
В	Exercice 02: ft_list_push_front	8
VI	Exercice 03: ft_list_size	9
VII	Exercice 04: ft_list_last	10
VIII Ex	ercice 05: ft_list_push_params	11
IX Exe	rcice 06: ft_list_clear	12
Икс	Exercice 07: ft_list_at	13
ΧI	Exercice 08: ft_list_reverse	14
XII	Exercice 09: ft_list_foreach	15
XIII Ex	ercice 10: ft_list_foreach_if	16
XIV Ex	tercice 11: ft_list_ фи-й	17
XV Ex	ercice 12: ft_list_remove_if	18
XVI Ex	tercice 13: ft_list_merge	19
XVII E	xercice 14: ft_list_sort	20
XVIII E	exercice 15: ft_list_reverse_fun	21
XIX Ex	ercice 16: ft_sorted_list_insert	22
XX Fv	ercice 17: ft sorted list merge	23

## Инструкции

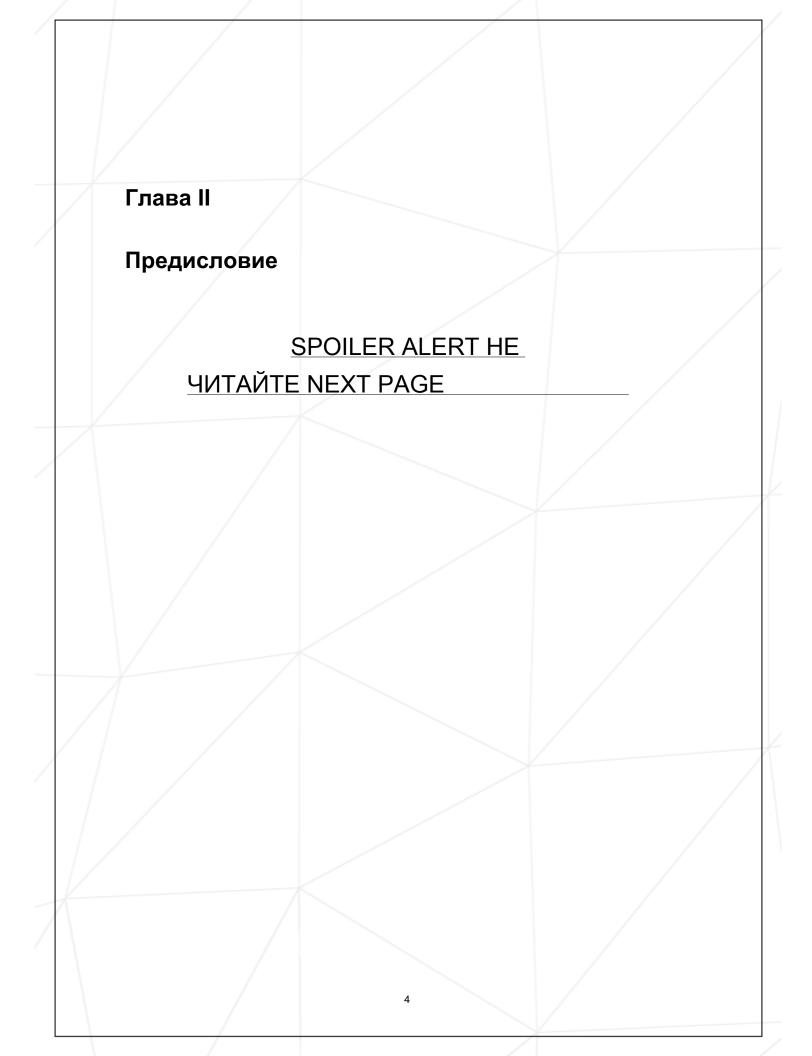
#### Глава I

- Только эта страница будет служить в качестве ссылки: не доверять слухам.
- Осторожно! Этот документ может потенциально изменить до часа до submis- Сионе.
- Убедитесь, что у вас есть соответствующие разрешения на вашем фи ле и каталогах.
- Вы должны следовать процедурам представления для каждого упражнения.
- Ваши упражнения будут проверены и рассортированы вашими одноклассниками.
- Кроме того, ваши упражнения будут проверены и классифицированы по программе под названием Moulinette.
- Мoulinette очень дотошный и строгий в своей оценке вашей работы. Она полностью автоматизирована и не существует никакого
  способа вести с ним переговоры. Так что если вы хотите, чтобы избежать неприятных сюрпризов, быть тщательным, насколько это
  возможно.
- Moulinette не очень открытым. Он не будет пытаться понять ваш код, если он не уважает нормы. Moulinette опирается на программу под названием Norminator чтобы проверить, если ваш фи ле уважать нормы. TL; DR: это было бы идиотизмом представить часть работы, которая не проходит Norminator «S проверка.
- Эти упражнения тщательно выложены порядка ди FFI трудности от простого к сложному. Мы не будет принимать во внимание успешно завершена сложнее упражнение, если проще один не вполне функциональны.
- Использование запрещенной функции считается мошенничеством. Мошенники получить 42, и этот сорт не является предметом переговоров.
- Если ft\_putchar () является авторизированным функция, мы будем компилировать код с нашей ft\_putchar.c.
- Вы должны будете только представить основную функцию (), если мы просим программу.

- Moulinette компилирует с этим фл флажков: -Wall -Wextra -Werror и использования НКА.
- Если ваша программа не компилируется, вы получите 0.
- Вы не можете оставить никакого дополнительные фи ль в каталоге, чем то Специфическое едь в этой теме.
- Есть вопрос? Попросите ваш партнёр по правой руке. В противном случае, попытать пэром на левой стороне.
- Ваш справочник называется Google / человек / Интернет / ....
- Проверьте «С» Piscine часть форума в интрасети.
- Изучите примеры тщательно. Они очень хорошо могли бы назвать подробности, которые явно не упомянутые в этой теме ...
- По Одина, Тор! Используй свой мозг!!!
- Для следующих упражнений, вы должны использовать следующую структуру:



- Вы должны включить эту структуру в ФИ ле ft\_list.h и представить его для каждого упражнения.
- Из физических упражнений 01 и далее, мы будем использовать наш ft\_create\_elem, поэтому принимать меры (это может быть полезно иметь свой прототип в фи ле ft\_list.h ...).



#### Вы были предупреждены.

- В Звездные войны, Dark Vador Отец Люка.
- В Обычные подозреваемые, Вербальные является Кайзер Soze.
- В Бойцовский клуб, Тайлер Дерден и рассказчик тот же человек.
- В Шестой Сенс, Брюс Уиллис мертв с самого начала.
- В Другие, жители дома привидение и наоборот.
- В Бэмби, мать Бэмби умирает.
- В Деревня, монстры жители и фильм на самом деле происходит в наше время.
- В Гарри Поттер, Дамблдор умирает.
- В Планета обезьян, фильм происходит на земле.
- В Игра престолов, Робб Старк и Джо Ф.Ф. Рей Баратеон умереть в день их свадьбы.
- В Сумерки, Вампиры светить под солнцем.
- В Stargate SG-1, сезон 1, эпизод 18, О'Нилл и Картер в Антарктиде.
- В Восстание темного рыцаря, Миранда Тейт Талия Алгул.
- В Super Mario Bros, Принцесса в другом замке.

## Глава III

# Exercice 00: ft\_create\_elem

	Exercice: 00	
	ft_create_elem	
Пошаговая в каталоге: <i>бывший</i> 00 / Файлы для включения в: ft_create_elem.c,		
ft_list.h		
Допустимые функции: та	Hoc	
Примечания: н /		

- · Создание функции ft\_create\_elem которая создает новый элемент t\_list тип.
- Он должен назначить данные к данному аргументу и следующий в NULL.
- Вот как это должно быть прототип:

\_list

\* ft\_create\_elem ( недействительным \* данные)

## Глава IV

# Exercice 01: ft\_list\_push\_back

3	Exercice: 01	
	ft_list_push_back	
Пошаговая в каталоге: <i>бывший</i> 01 / Файлы для включения в: ft_list_push_back.c,		
ft_list.h		
Допустимые функции: ft_create_elem		
Примечания: н /		

- Создание функции ft\_list\_push\_back который добавляет новый элемент t\_list введите в конце списка.
- Он должен назначить данные к данному аргументу.
- При необходимости, он будет обновлять указатель на начало списка.
- Вот как это должно быть прототип:

недействительным ft\_list\_push\_back (t\_list \*\* begin\_list, недействительным \* данные);

#### Глава V

# **Exercice 02: ft\_list\_push\_front**

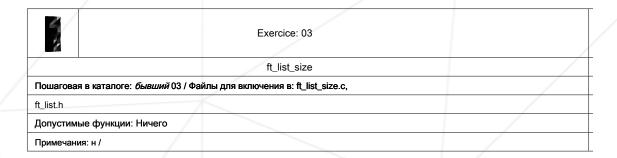
	Exercice: 02	
	ft_list_push_front	
Пошаговая в каталоге: бывший 02 / Файлы для включения в: ft_list_push_front.c,		
ft_list.h		
Допустимые функции: ft_create_elem		
Примечания: н /		

- Создание функции ft\_list\_push\_front который добавляет новый элемент типа t\_list к началу списка.
- Он должен назначить данные к данному аргументу.
- При необходимости, он будет обновлять указатель на начало списка.
- Вот как это должно быть прототип:

недействительным ft\_list\_push\_front (t\_list \*\* begin\_list, недействительным \* данные);

### Глава VI

## Exercice 03: ft\_list\_size

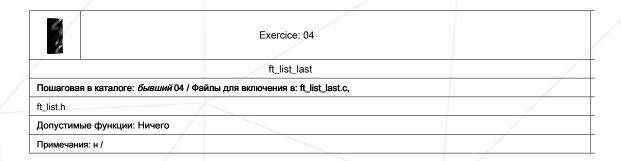


- Создание функции ft\_list\_size которая возвращает количество элементов в списке.
- Вот как это должно быть прототип:

MHT ft\_list\_size (t\_list \* begin\_list);

### Глава VII

## **Exercice 04: ft\_list\_last**



- Создание функции ft\_list\_last которая возвращает последний элемент списка.
- Вот как это должно быть прототип:

t\_list \* ft\_list\_last (t\_list \* begin\_list);

#### Глава VIII

# **Exercice 05: ft\_list\_push\_params**

	Exercice: 05	
	ft_list_push_params	
Пошаговая в каталоге: <i>бывший</i> 05 / Файлы для включения в: ft_list_push_params.c,		
ft_list.h		
Допустимые функции: ft_create_elem		
Примечания: н /		/

- Создание функции ft\_list\_push\_params который создает новый список, который включает в себя аргументы командной строки.
- Первый аргумент должен быть в конце списка.
- -адрес первого Линка фи в списке возвращается.
- Вот как это должно быть прототип:

t\_list \* ft\_list\_push\_params ( ИНТ переменный ток, голец \*\* средний);

## Глава IX

# Exercice 06: ft\_list\_clear

9	Exercice: 06	
/	ft_list_clear	/
Пошаговая в каталоге: б	ывший 06 / Файлы для включения в: ft_list_clear.c,	
ft_list.h		
Допустимые функции: свобод	но	
Примечания: н /		

- Создание функции ft\_list\_clear который удаляет все ссылки из списка.
- Затем он будет присвоить указатель в списке на нулевое значение.
- Вот как это должно быть прототип:

недействительным ft\_list\_clear (t\_list \*\* begin\_list);

## Глава Х

# Exercice 07: ft\_list\_at

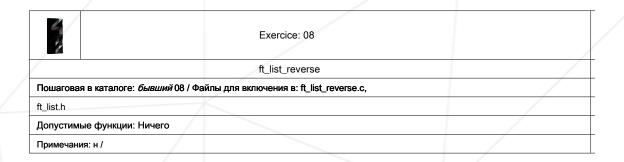
	Exercice: 07	
/	ft_list_at	
Пошаговая в каталоге: быв	ший 07 / Файлы для включения в: ft_list_at.c,	
ft_list.h		
Допустимые функции: Нич	чего	
Примечания: н /		

- Создание функции ft\_list\_at которая возвращает Nth элемент списка.
- В случае ошибки, она должна возвращать нулевой указатель.
- Вот как это должно быть прототип:

t\_list \* ft\_list\_at (t\_list \* begin\_list, неподписанных INT NBR);

#### Глава XI

## Exercice 08: ft\_list\_reverse

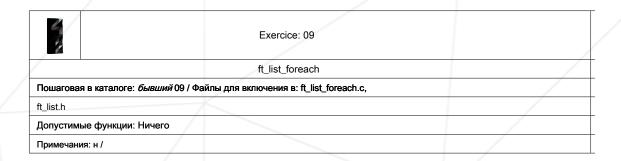


- Создание функции ft\_list\_reverse который меняет порядок элементов а список. Вы можете использовать только указатели, связанные ст и далее.
- Вот как это должно быть прототип:

недействительным ft\_list\_reverse (t\_list \*\* begin\_list);

#### Глава XII

## Exercice 09: ft\_list\_foreach



- Создание функции ft\_list\_foreach который применяет функцию, заданную в качестве аргумента к информации внутри каждой из ссылок списка.
- Вот как это должно быть прототип:

недействительным (\* e) ( медействительным (\* e) ( медействительным (\* e) (

• Функция указываемого е будет использоваться следующим образом:

( \* e) (list\_ptr -> данные);

#### Глава XIII

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



- Создание функции ft\_list\_foreach\_if который применяет функцию, заданную в качестве аргумента к информации, содержащейся в некоторых звеньях списка. Ссылка информации, а также сравнительная функция должна позволить нам выбрать правильные ссылки из списка: те, которые «равны» в справочной информации.
- Вот как это должно быть прототип:

elictorreльным ft\_list\_foreact\_if (t\_list \* begin\_list, надрайствительным (\* e) ( надрайствительным \*), надрайствительным \* data\_ref, MHT ( \* CMP) (

• Функции указываемого е и СМР будет использоваться следующим образом:

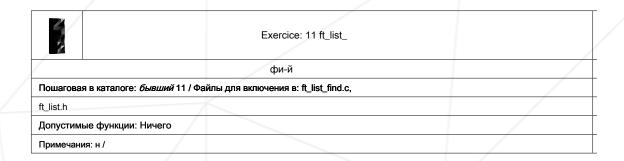
(\*e) (list\_ptr -> данные); (\*CMP) (list\_ptr -> Данные, data\_ref);



Например, функция СМР может быть ft\_strcmp ...

#### Глава XIV

## Exercice 11: ft\_list\_ фи-й

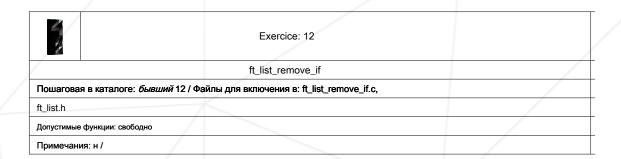


- Создание функции ft\_list\_find который возвращает адрес ссылки первой, чьи данные «равно» к справочным данным.
- Вот как это должно быть прототип:

t\_list \* ft\_list\_find (t\_list \* begin\_list, недействительным \* data\_ref, ИНТ ( \* СМР) ());

#### Глава XV

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



- Создание функции ft\_list\_remove\_if который стирает о далее список всех элементов, чьи данные «равны» справочным данным.
- Вот как это должно быть прототип:

недействительным ft\_list\_remove\_if (t\_list \*\* begin\_list, недействительным \* data\_ref, ИНТ ( \* CMP) ());

## Глава XVI

# Exercice 13: ft\_list\_merge

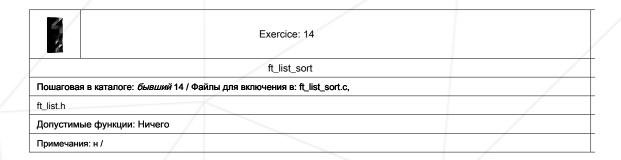
	Exercice: 13	
	ft_list_merge	
Пошаговая в каталоге: <i>бывший</i> 13 / Файлы для включения в: ft_list_merge.c,		
ft_list.h		
Допустимые функции: Н	ичего	
Примечания: н /		

- · Создание функции ft\_list\_merge который размещает элементы списка begin2 в конце списка в другой begin1.
- Создание элемента не разрешается.
- Вот как это должно быть прототип:

недействительным ft\_list\_merge (t\_list \*\* begin\_list1, t\_list \* begin\_list2)

#### Глава XVII

## Exercice 14: ft\_list\_sort



- Создание функции ft\_list\_sort который сортирует содержимое списка в порядке возрастания путем сравнения двух звеньев, благодаря функции, которая может сравнить данные, содержащиеся в этих двух ссылок.
- Вот как это должно быть прототип:

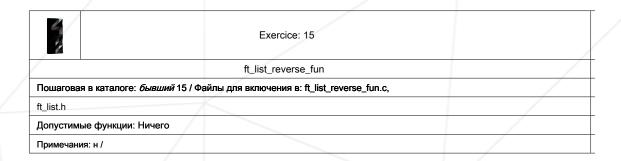
недействительным ft\_list\_sort (t\_list \*\* begin\_list, ИНТ ( \* СМР) ());



La fonction ЧМК pourrait être пар Exemple ft\_strcmp.

#### Глава XVIII

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

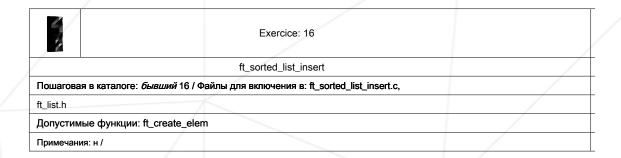


- Создание функции ft\_list\_reverse\_fun который меняет порядок элементов списка. Вы можете использовать только указатели, связанные ст и далее.
- Вот как это должно быть прототип:

недействительным ft\_list\_reverse\_fun (t\_list \* begin\_list);

#### Глава XIX

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



- Создание функции ft\_sorted\_list\_insert которая создает новый элемент и вставляет его в список, отсортированный так что остается sortend в порядке возрастания.
- Вот как это должно быть прототип:

медействительным ft\_sorted\_list\_insert (t\_list \*\* begin\_list, недействительным \* данные, ИНТ ( \* CMP) ());

## Глава XX

# Exercice 17: ft\_sorted\_list\_merge

	Exercice: 17	
	ft_sorted_list_merge	
Пошаговая в каталоге: бы		
ft_list.h		
Допустимые функции: Н	ичего	
Примечания: н /		

- Создание функции ft\_sorted\_list\_merge который объединяет в себе элементы отсортированного списка begin2 в другом отсортированном списке begin1, так что begin1 остается отсортирован в порядке возрастания.
- Вот как это должно быть прототип:

недействительным ft\_sorted\_list\_merge (t\_list \*\* begin\_list1, t\_list \* begin\_list2, ИНТ ( \* СМР) ());