



Piscine Reloaded

Приятно вернуться.

Краткое содержание:

Бассейн был хорош, но время прошло. Эта серия упражнений поможет вам вспомнить все основы, которые вы изучили во время бассейна. Функции, циклы, указатели, структур, давайте вместе вспомним синтаксические и семантические основы

Содержание

Я	Предисловие	2
II	Введение	3
III	Общие правила	4
IV	Упражнение 00: О да, еще...	6
V	Упражнение 01: Z	8
VI	Упражнение 02: чистота	9
VII	Упражнение 03: find_sh	10
VIII	Упражнение 04: MAK	11
IX	Упражнение 05: Можете ли вы это создать?	12
X	Упражнение 06: ft_print_alphabet	13
XI	Упражнение 07: ft_print_numbers	14
XII	Упражнение 03: ft_is_negative	15
XIII	Упражнение 09: ft_ft	16
XIV	Упражнение 10: ft_swap	17
XV	Упражнение 11: ft_div_mod	18
XVI	Упражнение 12: ft_iterative_factorial	19
XVII	Упражнение 13: ft_recursive_factorial	20
XVIII	Упражнение 14: ft_sqrt	21
XIX	Упражнение 15: ft_putstr	22
XX	Упражнение 16: ft_strlen	23
XXI	Упражнение 17: ft_strcmp	24
XXII	Упражнение 18: ft_print_params	25
XXIII	Упражнение 19: ft_sort_params	26

XXIV Упражнение 20: ft_strdup	27
XXV Упражнение 21: ft_range	28
XXVI Упражнение 22: ft_abs.h	29
XXVII Упражнение 23: ft_point.h	30
XXVIII Упражнение 24: Makefile	31
XXIX Упражнение 25: ft_foreach	32
XXX Упражнение 26: ft_count_if	33
XXXI Упражнение 27: display_file	34
XXXII Проверка и коллегияльная оценка	35

Глава I

Предисловие

Эдвард Джозеф Сноуден (родился 21 июня 1983 года) — американский специалист по компьютерам, бывший сотрудник Центрального разведывательного управления (ЦРУ) и бывший подрядчик правительства США, который в 2013 году без разрешения скопировал и раскрыл секретную информацию Агентства национальной безопасности (АНБ). Его разоблачения выявили многочисленные глобальные программы слежки, многие из которых осуществлялись АНБ и разведывательным альянсом «Пять глаз» при содействии телекоммуникационных компаний и правительств европейских стран.

В 2013 году Сноуден был нанят компанией Booz Allen Hamilton, подрядчиком АНБ, до этого работавшим в Dell и ЦРУ. 20 мая 2013 года Сноуден вылетел в Гонконг, оставив работу в учреждении АНБ на Гавайях, и в начале июня раскрыл тысячи секретных документов АНБ журналистам Гленну Гринвальду, Лоре Пойтрас и Юэну Макаскиллу. Сноуден привлёк международное внимание после того, как статьи, основанные на его материалах, появились в The Guardian и The Washington Post. Дальнейшие публикации были опубликованы и другими изданиями, включая Der Spiegel и The New York Times.

21 июня 2013 года Министерство юстиции США предъявило Сноудену обвинения по двум пунктам: нарушению Закона о шпионаже 1917 года и краже государственной собственности. Два дня спустя он прилетел в московский аэропорт Шереметьево, но российские власти отметили, что его американский паспорт аннулирован, и он был ограничен в своем пребывании в терминале аэропорта более месяца. В конечном итоге Россия предоставила ему право убежища сроком на один год, и неоднократное продление срока позволило ему оставаться в стране как минимум до 2020 года. По имеющимся данным, он проживает в Москве в неустановленном месте и продолжает искать убежища в других странах мира.

Сноудена, предмета споров, называли героем, разоблачителем, диссидентом, предателем и патриотом. Его разоблачения вызвали споры о массовой слежке, государственной тайне и балансе между национальной безопасностью и конфиденциальностью информации.

Есть очень хороший документальный фильм [онВОздесь](#).

Глава II

Введение

ThePiscine Reloadedэто лучшее из упражнений, которые вы делали во времяБассейн Счтобы напомнить вам все основыСязык программирования. Все упражнения необходимо выполнить полностью, чтобы открыть доступ к следующему проекту.

Если вы уже выполнили некоторые из этих упражнений во время курса Piscine C, мы настоятельно рекомендуем не поддаваться искушению восстановить свой старый код. Изучение программирования требует практики, и написание существующего кода не представляет интереса.

Глава 3


Общие правила

- Только эта страница будет служить справочной информацией; не доверяйте слухам.
- Убедитесь, что у вас есть соответствующие разрешения на доступ к файлам и каталогам.
- Вам необходимо следовать процедуре сдачи для каждого упражнения.
- Ваши упражнения будут проверены и оценены программой Moulinette.
- Moulinette очень дотошно и строго оценивает вашу работу. Он полностью автоматизирован, и с ним невозможно договориться. Поэтому, если хотите избежать неприятных сюрпризов, будьте максимально внимательны.
- Упражнения в Shell должны выполняться с помощью /bin/sh.
- Ты не можешь оставлять любой дополнительный файл в вашем каталоге, кроме указанных в теме.
- Есть вопрос? Задайте его собеседнику справа. Или обратитесь к собеседнику слева.
- Ваш справочник называется Google / человек / Интернет /
- Посетите форум в интранете и Slack.
- Внимательно изучите примеры. Они вполне могут потребовать деталей, которые явно не упомянуты в теме...
- Moulinette не очень открыт для нового. Он не будет пытаться понять ваш код, если он не соответствует Норме. Moulinette использует программу под названием Норминатор чтобы проверить, соответствуют ли ваши файлы норме. TL;DR: было бы глупо отправлять работу, которая не проходит Норминатор проверку.
- Использование запрещённой функции считается читерством. Читеры получают...42, и эта оценка не подлежит обсуждению.
- Вам нужно будет только отправить функцию main(), если мы запросим программа .

- Moulinette компилируется со следующими флагами: -Wall -Wextra -Werror и использует gcc.
- Если `ft_putchar()` является разрешенной функцией, мы скомпилируем ваш код с помощью нашей `ft_putchar.c`.
- Если ваша программа не компилируется, вы получите 0.
- Клянусь Одином, клянусь Тором! Пошевели мозгами!!!

Глава IV

Упражнение 00: О да, еще...

	Упражнение 00
О да, еще...	
Каталог сдачи: <i>бывший</i> 00/	
Файлы для сдачи: <i>exo</i> .tar	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте следующие файлы и каталоги. Сделайте всё необходимое, чтобы при использовании `ls -l` команды в вашем каталоге, вывод будет выглядеть так:

```
$> ls -l
всего 42
drwx--xr-x 2 login wheel XX 1 июня 20:47 test0
-rwx--xr-- 1 войти колесо 4 июн 1 21:46 test1
dr-x---r-- 2 войти колесо XX 1 июня 22:45 test2
-r-----r-- 2 войти колесо 1 июн 1 23:44 test3
-rw-r-----x 1 войти колесо 2 июн 1 23:43 test4
-r-----r-- 2 войти колесо 1 июн 1 23:44 test5
lrwxr-xr-x 1 колесо входа      5 июня 1 22:20 test6 -> test0
$>
```

- Что касается часов, то будет принято, если указан год, если дата упражнения (1 июня) устарела на шесть месяцев или более.
- Как только вы это сделаете, запустите `tar -cf exo.tar *` для создания файла для отправки.



«login» и «wheel» будут заменены на ваш логин и вашу группу соответственно.




У вас не получится использовать ту же строку «итого 42», как показано в примере.



Не беспокойтесь о том, что у вас вместо «XX».

Глава V

Упражнение 01: Z

	Упражнение 01
Только лучшие знают, как отображать Z	
Каталог сдачи: <i>бывший</i> 01/	
Файлы для сдачи: z	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте файл с именем `z`, который возвращает «Z», а затем новую строку, всякий раз, когда команда `cat` используется на нем.

```
?> cat z
Z
?>
```

Глава VI

Упражнение 02: чистота

	Упражнение 02
Каталог сдачи: <i>бывший02/</i>	
Файлы для сдачи: чистый	
Разрешенные функции: Никто	
Примечания :н/д	

- В файле под названием *чистый* поместите командную строку, которая будет искать все файлы - в текущем каталоге, а также в его подкаталогах - с именем, заканчивающимся на ~, или с именем, которое начинается и заканчивается на #
- Командная строка покажет и сотрет все найденные файлы.
- Разрешена только одна команда: никаких «;», «&&» и прочих махинаций.



человек находит

Глава VII

Упражнение 03: find_sh

	Упражнение 03
find_sh.sh	
Каталог сдачи: <i>бывший</i> 03/	
Файлы для сдачи: find_sh.sh	
Разрешенные функции: Никто	
Примечания :н/д	

- Напишите командную строку, которая ищет все имена файлов, заканчивающиеся на «.sh» (без кавычек), в текущем каталоге и всех его подкаталогах. Должны быть отображены только имена файлов без расширения «.sh».
- Пример вывода:

```
$>./find_sh.sh | cat -e
find_sh$
файл1$
файл2$
файл3$
$>
```

Глава VIII

Упражнение 04: МАК

	Упражнение 04
MAC.sh	
Каталог сдачи: <i>бывший</i> 04/	
Файлы для сдачи: MAC.sh	
Разрешенные функции: Никто	
Примечания :н/д	


- Напишите командную строку, которая выведет MAC-адреса вашего компьютера. После каждого адреса должен стоять перенос строки.



man ifconfig

Глава IX

Упражнение 05: Можете ли вы это создать?

	Упражнение 05
Вы можете это создать?	
Каталог сдачи: <i>бывший</i> 05/	
Файлы для сдачи: "\?\$*'KwaMe'*\$?\\"	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте файл, содержащий только «42», и НИЧЕГО больше.

- Его название будет:


```
"\?$*'KwaMe'*$?\\"
```

- Пример :

```
$>ls -lRa *waM* | cat -e  
- rw--xr-- 1 75355 32015 2 окт. 2 12:21 "\?$*'KwaMe'*$?\\"$ $>
```

Глава X

Упражнение 06: ft_print_alphabet


	Упражнение 06
ft_print_alphabet	
Каталог сдачи: <i>бывший</i> 06/	
Файлы для сдачи: ft_print_alphabet.c	
Разрешенные функции: ft_putchar	
Примечания :н/д	

- Создайте функцию, которая отображает алфавит строчными буквами в одной строке в порядке возрастания, начиная с буквы «а».
- Вот как это должно быть прототипировано:

```
пустотаft_print_alphabet(пустота);
```

Глава XI

Упражнение 07: ft_print_numbers


	Упражнение 07
ft_print_numbers	
Каталог сдачи: <i>бывший</i> 07/	
Файлы для сдачи: ft_print_numbers.c	
Разрешенные функции: ft_putchar	
Примечания :н/д	

- Создайте функцию, которая отображает все цифры в одной строке в порядке возрастания.
- Вот как это должно быть прототипировано:

```
пустотаft_print_numbers(пустота);
```


Глава XII

Упражнение 03: ft_is_negative

	Упражнение 08
ft_is_negative	
Каталог сдачи: <i>бывший</i> 08/	
Файлы для сдачи: ft_is_negative.c	
Разрешенные функции: ft_putchar	
Примечания :н/д	

- Создайте функцию, которая выводит «N» или «P» в зависимости от знака целого числа, указанного в качестве параметра. Если отрицательно, вывести «N». Если положительно или равно нулю, вывести «P».
- Вот как это должно быть прототипировано:

```
пустотаft_is_negative(интн);
```

Глава XIII

Упражнение 09: ft_ft

	Упражнение 09
фут_фут	
Каталог сдачи: <i>бывший</i> 09/	
Файлы для сдачи: ft_ft.c	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте функцию, которая принимает указатель на int в качестве параметра и присваивает этому int значение «42».
- Вот как это должно быть прототипировано:

```
пустотафут_фут(инт*нбр);
```

Глава XIV

Упражнение 10: ft_swap

	Упражнение 10
ft_swap	
Каталог сдачи: <i>бывший</i> 10/	
Файлы для сдачи: ft_swap.c	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте функцию, которая меняет местами значения двух целых чисел, адреса которых введены в качестве параметров.
- Вот как это должно быть прототипировано:

```
пустота ft_swap(инт*а,инт*б);
```

Глава XV

Упражнение 11: ft_div_mod

	Упражнение 11
ft_div_mod	
Каталог сдачи: <i>бывший</i> 11/	
Файлы для сдачи: ft_div_mod.c	
Разрешенные функции: Никто	
Примечания :н/д	


- Создать функцию ft_div_mod прототипирован так:

```
пустота ft_div_mod(инта, интб, инт*див, инт*мод);
```

- Эта функция делит параметры *акби* и сохраняет результат в *инт*, на который указывает *див*. Он также сохраняет остаток деления *акбв* *инт*, на который указывает *мод*.

Глава XVI

Упражнение 12: ft_iterative_factorial

	Упражнение 12
ft_iterative_factorial	
Каталог сдачи: <i>бывший</i> 12/	
Файлы для сдачи: ft_iterative_factorial.c	
Разрешенные функции: Никто	
Примечания :н/д	


- Создайте итеративную функцию, возвращающую число. Это число является результатом факториала числа, заданного в качестве параметра.
- В случае ошибки функция должна вернуть 0.
- Вот как это должно быть прототипировано:

```
int ft_iterative_factorial(int n);
```

- Ваша функция должна вернуть результат менее чем за две секунды.

Глава XVII

Упражнение 13: ft_recursive_factorial

	Упражнение 13
ft_recursive_factorial	
Каталог сдачи: <i>бывший</i> 13/	
Файлы для сдачи: ft_recursive_factorial.c	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте рекурсивную функцию, которая возвращает факториал числа, заданного в качестве параметра.
- В случае ошибки функция должна вернуть 0.
- Вот как это должно быть прототипировано:

```
интft_recursive_factorial(интнб);
```

Глава XVIII

Упражнение 14: ft_sqrt

	Упражнение 14
ft_sqrt	
Каталог сдачи: <i>бывший</i> 14/	
Файлы для сдачи: ft_sqrt.c	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте функцию, которая возвращает квадратный корень числа (если он существует) или 0, если квадратный корень является иррациональным числом.
- Вот как это должно быть прототипировано:

```
интft_sqrt(интнб);
```

- Ваша функция должна вернуть результат менее чем за две секунды.

Глава XIX

Упражнение 15: ft_putstr


	Упражнение 15
ft_putstr	
Каталог сдачи: <i>бывший</i> 15/	
Файлы для сдачи: ft_putstr.c	
Разрешенные функции: ft_putchar	
Примечания :н/д	

- Создайте функцию, которая выводит строку символов на стандартный вывод.
- Вот как это должно быть прототипировано:

```
пустота ft_putstr(чар*ул);
```


Глава XX

Упражнение 16: ft_strlen

	Упражнение 16
ft_strlen	
Каталог сдачи: <i>бывший</i> 16/	
Файлы для сдачи: ft_strlen.c	
Разрешенные функции: Никто	
Примечания : н/д	

- Воспроизвести поведение функции `strlen` (`man strlen`).
- Вот как это должно быть прототипировано:

```
int ft_strlen(char* ул);
```

Глава XXI

Упражнение 17: ft_strcmp


	Упражнение 17
ft_strcmp	
Каталог сдачи: <i>бывший17/</i>	
Файлы для сдачи: ft_strcmp.c	
Разрешенные функции: Никто	
Примечания :н/д	

- Воспроизвести поведение функции strcmp (man strcmp).
- Вот как это должно быть прототипировано:

```
интft_strcmp(чар*c1,чар*c2);
```

Глава XXII

Упражнение 18: ft_print_params


	Упражнение 18
ft_print_params	
Каталог сдачи: <i>бывший</i> 18/	
Файлы для сдачи: ft_print_params.c	
Разрешенные функции: ft_putchar	
Примечания :н/д	

- Мы имеем дело с программой здесь, следовательно, у вас должна быть функция основной в вашем .c файле.
- Создайте программу, которая отображает заданные аргументы.
- Пример :

```
$> ./a.out тест1 тест2 тест3
тест1
тест2
тест3
$>
```

Глава XXIII

Упражнение 19: ft_sort_params

	Упражнение 19
ft_sort_params	
Каталог сдачи: <i>бывший</i> 19/	
Файлы для сдачи: ft_sort_params.c	
Разрешенные функции: ft_putchar	
Примечания :н/д	

- Мы имеем дело с программой здесь, следовательно, у вас должна быть функция основной в вашем .c file.
- Создайте программу, которая отображает заданные аргументы, отсортированные по порядку ASCII.
- Он должен отображать все аргументы, за исключением argv[0].
- Все аргументы должны иметь свою собственную линию.

Глава XXIV

Упражнение 20: ft_strdup

	Упражнение 20
ft_strdup	
Каталог сдачи: <i>бывший20/</i>	
Файлы для сдачи: ft_strdup.c	
Разрешенные функции: malloc	
Примечания :н/д	

- Воспроизвести поведение функции strdup (man strdup).
- Вот как это должно быть прототипировано:

```
char *ft_strdup(char *источник);
```

Глава XXV

Упражнение 21: ft_range

	Упражнение 21
ft_range	
Каталог сдачи: <i>бывший</i> 21/	
Файлы для сдачи: ft_range.c	
Разрешенные функции: malloc	
Примечания :н/д	


- Создать функцию `ft_range` который возвращает массив целые. Этот `инт` массив должен содержать все значения между `мин` и `макс`.
- `Мин`. включено - `макс` исключено.
- Вот как это должно быть прототипировано:

```
инт *ft_range(инт мин, инт макс);
```

- Если `мин` значение больше или равно `макс`-значения, должен быть возвращен нулевой указатель.

Глава XXVI

Упражнение 22: ft_abs.h

	Упражнение 22
ft_abs.h	
Каталог сдачи: <i>бывший22/</i>	
Файлы для сдачи: ft_abs.h	
Разрешенные функции: Никто	
Примечания :н/д	

- Создайте макрос ABS, который заменяет свой аргумент его абсолютным значением:

```
# определить ABS(Значение)
```



Вас просят сделать что-то, что обычно запрещено, и это единственный случай, когда мы это разрешим.

Норм,

Глава XXVII

Упражнение 23: ft_point.h

	Упражнение 23
ft_point.h	
Каталог сдачи: <i>бывший23/</i>	
Файлы для сдачи: ft_point.h	
Разрешенные функции: Никто	
Примечания :н/д	

- Создать файл ft_point.h, который скомпилирует следующий основной файл:

```
# включить "ft_point.h"

пустота set_point(t_point* точка) {

    точка->x= 42;
    точка->y= 21;
}

ИНТ    основной(пустота)
{
    t_point    точка;

    set_point(&точка);
    возвращаться(0);
}
```


Глава XXVIII

Упражнение 24: Makefile

	Упражнение 24
Makefile	
Каталог сдачи: <i>бывший24/</i>	
Файлы для сдачи: Makefile	
Разрешенные функции: Никто	
Примечания :н/д	


- Создайте Makefile который скомпилирует ваш `libft.a`.
- The Makefile получит исходные файлы из каталога `"srcs"`.
- The Makefile получит свои заголовочные файлы из каталога `«includes»`.
- Библиотека будет лежать в основе упражнения.
- The Makefile также должны выполнять следующие правила: чистый, чистый и повторно а также все.
- `fclean` выполняет эквивалент `make clean`, а также стирает двоичный файл, созданный во время `make`. `повторно` выполняет эквивалент команды `make fclean`, за которой следует `make`.
- Мы только скачаем ваш Makefile и протестируем его с нашими файлами. Для этого упражнения необходимо обработать только следующие 5 обязательных функций вашей библиотеки: (`ft_putchar`, `ft_putstr`, `ft_strcmp`, `ft_strlen` и `ft_swap`).



Остерегайтесь джокеров!

Глава XXIX

Упражнение 25: ft_foreach

	Упражнение 25
ft_foreach	
Каталог сдачи: <i>бывший25/</i>	
Файлы для сдачи: ft_foreach.c	
Разрешенные функции: Никто	
Примечания :н/д	

- Создать функцию `ft_foreach` Эта функция, к заданному массиву целых чисел, применяет функцию ко всем элементам массива. Эта функция будет применена в порядке, соответствующем порядку элементов массива.
- Вот как должна быть прототипирована функция:


```
пустота ft_foreach(инт* вкладка, инт длина, пустота (*ф)(инт));
```

- Например, функция `ft_foreach` Для отображения всех целых чисел массива можно вызвать его следующим образом:

```
ft_foreach(tab, 1337, &ft_putnbr);
```

Глава XXX

Упражнение 26: ft_count_if

	Упражнение 26
ft_count_if	
Каталог сдачи: <i>бывший26/</i>	
Файлы для сдачи: ft_count_if.c	
Разрешенные функции: Никто	
Примечания :н/д	

- Создать функцию `ft_count_if` который вернет количество элементов массива, возвращающих 1, переданное в функцию `ф`.
- Вот как должна быть прототипирована функция:

```
инт ft_count_if(чар**вкладка, инт(*ф)(чар*));
```

- Массив будет разделен 0.

Глава XXXI

Упражнение 27: display_file

	Упражнение 27
display_file	
Каталог сдачи: <i>бывший27/</i>	
Файлы для сдачи: Makefile и файлы, необходимые для вашей программы	
Разрешенные функции: закрыть, открыть, прочитать, написать	
Примечания :н/д	

- Создайте программа называется `ft_display_file` которая выводит на стандартный вывод только содержимое файла, указанного в качестве аргумента.
- Каталог отправки должен иметь `Makefile` со следующими правилами: все, чисто, чисто. Двоичный файл будет называться `ft_display_file`.
- `malloc` функция запрещена. Выполнить это упражнение можно, только объявив массив фиксированного размера.
- Все файлы, указанные в качестве аргументов, будут действительны.
- Сообщения об ошибках должны отображаться на зарезервированном для них выходе.

```
$> ./ft_display_file
Имя файла отсутствует.
$> ./ft_display_file Makefile
* содержимое файла Makefile*
$> ./ft_display_file Makefile display_file.c Слишком
много аргументов.
$>
```

Глава XXXII

Rendu et peer-évaluation

Свидание с вашим трудом на вашем складеГИТкак привычка. Сеул, который сейчас находится в своем складе, очень цененГлубокие мысли.

Je vous Rappelle Qu'Exceptionnellement, этот проект будет уникальным по своей сути.Глубокие мысли. Это не аура, не требующая коллегальной оценки.

Единственное примечание, что приемлемо для реализации этого проекта, составляет 100%. Си Глубокие мыслиvous attribue une note inférieure, c'est que vous avez échoué et que vous devez reessayer de faire mieux en cliquant sur le boutonПовторить попыткуde votre passage de projet.

Приятного мужества всем и всем, кто хочет стать автором!