



ft_printf

Потому что `ft_putnbr()` и `ft_putstr()`
недостаточно.

Резюме:

*Цель этого проекта довольно проста. Вы перекодируете функцию `printf()`.
В основном вы узнаете об использовании переменного числа аргументов. Насколько
это круто?*

На самом деле это очень круто :)

Версия: 9.2

Содержание

I	Введение	2
II	Общие инструкции	3
III	Обязательная часть	5
IV	Бонусная часть	7
V	Представление и экспертная оценка	8

Глава I Введение

Вы познакомитесь с популярной и универсальной функцией языка Си: `printf()`. Это упражнение - отличная возможность улучшить свои навыки программирования. Оно имеет умеренную сложность.

Вы познакомитесь с **переменными функциями** в Си.

Ключом к успеху `ft_printf` является хорошо структурированный и расширяемый код.



После выполнения этого задания вам будет разрешено добавить `ft_printf()` в `libft`, чтобы вы могли использовать его в своих школьных проектах на Си.

Глава II

Общие инструкции

- Ваш проект должен быть написан на языке C.
- Ваш проект должен быть написан в соответствии с Нормой. Если у вас есть бонусные файлы/функции, они включаются в проверку нормы, и вы получите 0, если внутри есть ошибка нормы.
- Ваши функции не должны завершаться неожиданно (segmentation fault, bus error, double free, etc), кроме неопределенного поведения. Если это произойдет, ваш проект будет считаться не функциональным и получит 0 баллов во время оценки.
- Все выделенное пространство памяти должно быть надлежащим образом освобождено, когда это необходимо. Утечки не допускаются.
- Если тема требует этого, вы должны предоставить Makefile, который скомпилирует ваши исходные файлы до требуемого результата с флагами -Wall, -Wextra и -Werror, использовать cc, и ваш Makefile не должен перелинковываться.
- Ваш Makefile должен, по крайней мере, содержать правила \$(NAME), all, clean, fclean и re.
- Чтобы внести бонусы в свой проект, вы должны включить в Makefile правило bonus, которое добавит все различные заголовки, либрейки или функции, запрещенные в основной части проекта. Бонусы должны находиться в другом файле _bonus.{c/h}, если в теме не указано ничего другого. Оценка обязательной и бонусной частей производится отдельно.
- Если ваш проект позволяет использовать свою libft, вы должны скопировать ее исходные тексты и связанный с ней Makefile в папку libft с ее связанным Makefile. Makefile вашего проекта должен скомпилировать библиотеку, используя ее Makefile, а затем скомпилировать проект.
- Мы рекомендуем вам создавать тестовые программы для вашего проекта, даже если эта работа **не будет представлена и не будет оценена**. Это даст вам возможность легко проверить свою работу и работу ваших коллег. Эти тесты будут особенно полезны во время защиты. Действительно, во время защиты вы можете использовать свои тесты и/или тесты коллеги, которого вы оцениваете.
- Отправьте свою работу в назначенный вам git-репозиторий. Оцениваться

будет только работа в git-репозитории. Если Deerpthought назначит оценку вашей работы, это будет сделано

ft_printfПотому что `ft_putnbr()` и `ft_putstr()` недостаточно

после оценки работы сверстниками. Если во время оценивания Deerpthought в каком-либо разделе вашей работы будет допущена ошибка, оценка будет остановлена.

Глава III

Обязательная часть

Название программы	libftprintf.a
Сдать файлы	Makefile, *.h, */*.h, *.c, */*.c
Makefile	NAME, all, clean, fclean, re
Внешние функции.	malloc, free, write, va_start, va_arg, va_copy, va_end
Либфт уполномочен	Да
Описание	Напишите библиотеку, которая содержит ft_printf(), а функция, которая будет имитировать оригинальную функцию printf()

Вы должны перекодировать функцию printf() из libc.

Прототипом ft_printf() является:

```
int    ft_printf(const char *, ...);
```

Вот требования:

- Не реализуйте управление буфером оригинального printf().
- Ваша функция должна обрабатывать следующие преобразования: cspdiuxX%
- Ваша функция будет сравниваться с оригинальной функцией printf().
- Вы должны использовать команду ag для создания библиотеки. Использование команды libtool запрещено.
- Ваш libftprintf.a должен быть создан в корне вашего репозитория.

Вы должны реализовать следующие преобразования:

- %c Печатает один символ.
- %s Печатает строку (в соответствии с общепринятой конвенцией языка C).
- %p Аргумент указателя void * должен быть выведен в шестнадцатеричном формате.
- %d Выводит десятичное (основание 10) число.
- %i Печатает целое число по основанию 10.
- %u Выводит беззнаковое десятичное (основание 10) число.
- %x Выводит число в шестнадцатеричном (основание 16) строчном формате.
- %X Выводит число в шестнадцатеричном формате (основание 16) в верхнем регистре.
- %% Печатает знак процента.

Глава IV

Бонусная часть

Вам не обязательно выполнять все бонусы.

Список бонусов:

- Управляйте любой комбинацией следующих флагов: '-0.' и минимальную ширину поля для всех преобразований.
- Управляйте всеми следующими флагами: '# +' (Да, один из них - пробел)



Если вы планируете выполнить бонусную часть, продумайте реализацию ваших дополнительных функций с самого начала. Таким образом, вы избежите подводных камней наивного подхода.



Бонусная часть оценивается только в том случае, если обязательная часть выполнена безупречно. Совершенство означает, что обязательная часть выполнена полностью и работает без сбоев. Если вы не выполнили ВСЕ обязательные требования, ваша бонусная часть не будет оцениваться вообще.

Глава V

Представление и экспертная оценка

Сдайте задание в свой Git-репозиторий, как обычно. Во время защиты будет оцениваться только работа, находящаяся в вашем репозитории. Не стесняйтесь дважды проверять имена файлов, чтобы убедиться в их правильности.

После выполнения этого задания вам будет разрешено добавить свою `ft_printf()` в свой `libft`, чтобы вы могли использовать его в своих школьных проектах на Си.

