**Перевод задания "ft\_printf"**

**ft\_printf**

**Потому что ft\_putnbr() и ft\_putstr() недостаточно**

**Краткое описание**

Цель этого проекта довольно проста: вам нужно заново реализовать printf().

Вы изучите использование переменного числа аргументов. Разве это не круто? Это действительно полезный навык!

**Содержание**

1. Введение
2. Общие инструкции
3. Обязательная часть
4. Дополнительная часть (бонус)
5. Отправка работы и оценивание

**Глава I: Введение**

Вы познакомитесь с популярной и универсальной функцией C — printf().

Это упражнение является отличной возможностью улучшить ваши навыки программирования. Оно имеет средний уровень сложности.

Вы изучите работу с вариативными функциями в C.

Ключ к успешной реализации ft\_printf() — это хорошо структурированный и расширяемый код.

После успешного выполнения этого задания вам будет разрешено добавить ft\_printf() в вашу libft, чтобы использовать его в школьных C-проектах.

**Глава II: Общие инструкции**

* Ваш проект должен быть написан на C.
* Код должен соответствовать требованиям Norm (правила оформления кода).
* Все дополнительные файлы и функции, если они есть, должны соответствовать Norm, иначе работа будет оценена в 0.
* Ваши функции не должны приводить к неожиданным ошибкам (сегментационным ошибкам, ошибкам шины, двойному освобождению памяти и т. д.). Если это произойдет, работа будет считаться нефункциональной и получит 0.
* Все динамически выделенные участки памяти должны быть корректно освобождены. Утечки памяти не допускаются.
* Если в задании требуется, вы должны предоставить Makefile, который будет компилировать исходные файлы с флагами -Wall -Wextra -Werror, используя компилятор cc. Makefile **не должен выполнять повторную компоновку**.
* Ваш Makefile должен содержать как минимум следующие правила:
  + $(NAME)
  + all
  + clean
  + fclean
  + re
* Для выполнения бонусной части необходимо добавить правило bonus в Makefile.
* Если вам разрешено использовать libft, необходимо скопировать ее исходники и Makefile в папку libft. Ваш Makefile должен сначала компилировать libft, а затем ваш проект.
* Мы рекомендуем вам написать тестовые программы для проверки работы вашего ft\_printf(). Эти тесты не подлежат сдаче, но помогут вам во время защиты. Вы можете использовать свои тесты или тесты вашего сокурсника при проверке кода.
* Отправьте вашу работу в соответствующий Git-репозиторий. **Оценивается только код в репозитории**.

**Глава III: Обязательная часть**

**Описание проекта**

* **Название библиотеки**: libftprintf.a
* **Файлы для сдачи**: Makefile, заголовочные файлы (\*.h), исходные файлы (\*.c).
* **Используемые функции**:
  + malloc, free, write
  + va\_start, va\_arg, va\_copy, va\_end
* **Разрешено использование libft**
* **Описание**:
  + Вам необходимо реализовать библиотеку, содержащую ft\_printf(), которая будет повторять функциональность стандартного printf().
  + Прототип функции:

c

КопироватьРедактировать

int ft\_printf(const char \*, ...);

* + **Требования**:
    - Не реализовывать буферизацию, как в оригинальном printf().
    - Поддерживать следующие спецификаторы формата:
      * %c — печать одного символа.
      * %s — печать строки.
      * %p — печать указателя в шестнадцатеричном формате.
      * %d — десятичное (целое) число.
      * %i — целое число в десятичной системе.
      * %u — беззнаковое десятичное число.
      * %x — шестнадцатеричное число (нижний регистр).
      * %X — шестнадцатеричное число (верхний регистр).
      * %% — печать символа %.
    - Ваша функция будет сравниваться с оригинальной printf().
    - **Создавать библиотеку с помощью команды ar**. Использование libtool запрещено.
    - Файл libftprintf.a должен находиться в корневом каталоге вашего репозитория.

**Глава IV: Бонусная часть**

(Бонусная часть проверяется **только если обязательная часть выполнена без ошибок**)

Вы не обязаны выполнять все бонусные задания.

**Дополнительные возможности:**

* **Обработка флагов**: '-0.' и минимальной ширины поля для всех спецификаторов.
* **Поддержка всех следующих флагов**: '# + ' (да, один из них — это пробел).

💡 **Совет**: если планируете реализовывать бонусную часть, продумайте архитектуру кода заранее. Это поможет избежать проблем при добавлении новых возможностей.

**Важно**:

* Бонусная часть проверяется **только если обязательная часть реализована без ошибок**.
* Если обязательная часть содержит ошибки или работает некорректно, бонусная часть **не будет оцениваться**.

**Глава V: Отправка работы и оценивание**

* Выложите вашу работу в соответствующий Git-репозиторий. **Оценивается только код в репозитории**.
* Перед отправкой убедитесь, что все файлы имеют правильные имена.
* После успешной сдачи задания вам будет разрешено добавить ft\_printf() в вашу libft для использования в школьных C-проектах.