ft_printf project

- Функция должна называться ft_printf и иметь прототип сходный с оригинальным **int ft_printf (const char *format, . . .);**
  возвращает число выведенных символов или отрицательное значение в случае ошибки.

- buffer management???

- Спецификаторы **c s p**
  c - вывод символа (в качестве параметра подается одиночный символ или инт)**;**
  s - вывод строки символов (в качестве параметра подается только строка символов)**;**
  p - вывод адреса указателя (в качестве параметра подается адрес переменной для числа или символа. Для строки символов можно подать как просто строку так и её адрес)
  Например:
  char str[] = “12345” можно подать в printf как printf(“%p”, str) или printf(“%p”, &str)

- спецификаторы **d i o u x X** (https://ru.wikipedia.org/wiki/Printf)
  возможность использования с флагами **hh l ll**

  d, i - десятичное знаковое число, размер по умолчанию, sizeof(int). По умолчанию записывается с правым выравниванием, знак пишется только для отрицательных чисел. '%d' и '%i' ведут себя одинаково при выводе, но имеют разные значения при вводе с помощью функции scanf();

  o - восьмеричное беззнаковое число, размер по умолчанию sizeof(int);

  u - десятичное беззнаковое число, размер по умолчанию sizeof(int);

  x, X - шестнадцатеричное беззнаковое число, x использует маленькие буквы (abcdef), X большие (ABCDEF), размер по умолчанию sizeof(int);

- возможность использования (d i o u x X) с флагами **hh l ll**
  hh - Аргумент имеет тип int или unsigned int, но принудительно приводится к типу signed char или unsigned char, соответственно;

  l - long int или unsigned long int;

ll - long long int или unsigned long long int;

- **f** с флагами **l** или **L**
  f - числа с плавающей запятой. По умолчанию выводятся с точностью 6, если число по модулю меньше единицы, перед десятичной точкой пишется 0. Величины ±∞ представляются в форме [-]inf или [-]infinity, Величина Nan представляется как [-]nan или [-]nan(любой текст далее). Использование F выводит указанные величины заглавными буквами (-INF, NAN). Аргумент по умолчанию имеет размер double.

  L - значение аргумента изменяется с double на long double

  l - не нашел. Есть предположение, что этот флаг никак не меняет поведение функции применительно к спецификатору **f**

- **%%** - вывод знака %

- Поддержка флагов **# 0 - + пробел**

  # - Непосредственное расположение # перед спецификаторами преобразования **f (оставил только то, что есть в сабджекте, оригинал** https://cpp.com.ru/shildt_spr_po_c/08/0804.html**)**, означает что при выводе обязательно появится десятичная точка — даже если десятичных цифр нет. Если вы поставите # непосредственно перед x или X, то шестнадцатеричное число будет выведено с префиксом 0x. Если # будет непосредственно предшествовать спецификатору преобразования o, число будет выведено с ведущим нулем. К любым другим спецификаторам преобразования модификатор # применять нельзя.

  0 - дополнять поле до ширины, указанной в поле *ширина* управляющей последовательности, символом 0 (без этого знака дополняется пробелами). Используется для типов d, i, o, u, x, X, a, A, e, E, f, F, g, G. Для типов d, i, o, u, x, X, если *точность* указана, этот флаг игнорируется. Для остальных типов поведение не определено.

  - (знак минус) выводимое значение выравнивается по левому краю в пределах минимальной ширины поля (без знака “-” выравнивание по правому краю)

  + (знак плюс) всегда указывать знак (плюс или минус) для выводимого десятичного числового значения (без знака “+” только для отрицательных чисел)

  пробел - помещать перед результатом пробел, если первый символ

значения не знак (без знака вывод может начинаться с цифры). Символ **+** имеет больший приоритет, чем пробел. Используется только для десятичных числовых значений.

- Модификатор ширины - может быть десятичным числом или знаком “ * ” с соответствующим аргументом в списке аргументов.

- Модификатор точности.

- указывает на минимальное количество символов, которое должно появиться при обработке типов d, i, o, u, x, X;
- указывает на минимальное количество символов, которое должно появиться после десятичной запятой (точки) при обработке типа f
- максимальное число символов, которые будут выведены для типа s;
- Точность задаётся в виде точки с последующим десятичным числом или звёздочкой (*), если число или звёздочка отсутствует (присутствует только точка), то предполагается, что число равно нулю. Точка для указания точности используется даже в том случае, если при выводе чисел с плавающей запятой выводится запятая. Если после точки указан символ «звёздочка», то при обработке строки форматирования значение для поля читается из списка аргументов. (При этом, если символ звёздочка и в поле ширины и в поле точности, сначала указывается ширина, потом точность и лишь потом значение для вывода). Например, printf("%0*.*f", 8, 4, 2.5); выведет текст 002.5000.

Источники информации:

- по большей части взято из википедии https://ru.wikipedia.org/wiki/Printf
- про модификаторы * и # брал информацию отсюда https://cpp.com.ru/shildt_spr_po_c/08/0804.html