Задания к работе №2 по Фундаментальным алгоритмам.

1. Реализуйте функцию с переменным числом аргументов, определяющую для каждой из переданных десятичных дробей (в виде значения вещественного типа в диапазоне (0;1)) имеет ли она в системе счисления с переданным как параметр функции основанием конечное представление. Продемонстрируйте работу функции.
2. Реализовать следующие функции:
    a. void *memchr(const void *str, int c, size_t n) Выполняет поиск первого вхождения символа c (беззнаковый тип) в первых n байтах строки, на которую указывает аргумент str.
    b. int memcmp(const void *str1, const void *str2, size_t n) Сравнивает первые n байтов str1 и str2.
    c. void *memcpy(void *dest, const void *src, size_t n) Копирует n символов из src в dest.
    d. void *memset(void *str, int c, size_t n) Копирует символ c (беззнаковый тип) в первые n символов строки, на которую указывает аргумент str.
    e. char *strncat(char *dest, const char *src, size_t n) Добавляет строку, на которую указывает src, в конец строки, на которую указывает dest, длиной до n символов.
    f. char *strchr(const char *str, int c) Выполняет поиск первого вхождения символа c (беззнаковый тип) в строке, на которую указывает аргумент str.
    g. int strncmp(const char *str1, const char *str2, size_t n) Сравнивает не более первых n байтов str1 и str2.
    h. char *strncpy(char *dest, const char *src, size_t n) Копирует до n символов из строки, на которую указывает src, в dest.
    i. size_t strcspn(const char *str1, const char *str2)Вычисляет длину начального сегмента str1, который полностью состоит из символов, не входящих в str2.
    j. char *strerror(int errnum) Выполняет поиск во внутреннем массиве номера ошибки errnum и возвращает указатель на строку с сообщением об ошибке. Нужно объявить макросы, содержащие массивы сообщений об ошибке для операционных систем mac и linux. Описания ошибок есть в оригинальной библиотеке. Проверка текущей ОС осуществляется с помощью директив.
    k. size_t strlen(const char *str) Вычисляет длину строки str, не включая завершающий нулевой символ.
    l. char *strpbrk(const char *str1, const char *str2) Находит первый символ в строке str1, который соответствует любому символу, указанному в str2.
    m. char *strrchr(const char *str, int c) Выполняет поиск последнего вхождения символа c (беззнаковый тип) в строке, на которую указывает аргумент str.
    n. char *strstr(const char *haystack, const char *needle) Находит первое вхождение всей строки needle (не включая завершающий нулевой символ), которая появляется в строке haystack.

o. char *strtok(char *str, const char *delim) Разбивает строку str на ряд токенов, разделенных delim.

3. Реализуйте функции overfprintf и oversprintf, поведение которых схоже с поведением стандартных функций fprintf и sprintf, то есть эти функции имеют одинаковый прототип и логику работы, но в ваших функциях помимо стандартных флагов определены следующим образом дополнительные флаги:
   a. %Ro – печать в поток вывода целого числа типа int, записанного римскими цифрами;
   b. %Zr – печать в поток вывода цекендорфова представления целого числа типа unsigned int (коэффициенты 0 и 1 при числах Фибоначчи должны быть записаны от младшего к старшему слеванаправо с дополнительной единицей в конце записи, репрезентирующей окончание записи);
   c. %Cv печать целого числа типа int в системе счисления с заданным основанием (при обработке флага первым параметром функции, “снимаемым” со стека, является целое число типа int, вторым - основание целевой системы счисления в диапазоне [2..36] (при основании системы счисления, не входящем в диапазон, значение основания системы счисления устанавливается равным 10)); символы букв в результирующем строковом представлении целого числа должны быть записаны в нижнем регистре;
   d. %CV – аналогично флагу %Cv, при этом символы букв во входном строковом представлении целого числа должны быть записаны в верхнем регистре;
   e. %to – печать в поток вывода результата перевода целого числа, записанного в строковом представлении в системе счисления с заданным основанием в систему счисления с основанием 10 (при обработке флага первым параметром функции, “снимаемым” со стека, является строка, описываемая значением типа char *, вторым - основание исходной системы счисления в диапазоне [2..36] (при основании системы счисления, не входящем в диапазон, значение основания системы счисления устанавливается равным 10)); символы букв во входном строковом представлении целого числа должны быть записаны в нижнем регистре;
   f. %TO - аналогично флагу %to, при этом символы букв во входном строковом представлении целого числа должны быть записаны в верхнем регистре;
   g. %mi – печать дампа памяти (байты значения, записанные в системе счисления с основанием 2, в порядке нахождения в памяти “слева направо”; строковые представления байтов должны сепарироваться одним символом пробела) значения знакового целого 4-байтного числа;
   h. %mu – печать дампа памяти (байты значения, записанные в системе счисления с основанием 2, в порядке нахождения в памяти “слева направо”; строковые представления байтов должны сепарироваться одним символом пробела) значения беззнакового целого 4-байтного числа;
   i. %md – печать дампа памяти (байты значения, записанные в системе счисления с основанием 2, в порядке нахождения в памяти “слева направо”; строковые представления байтов должны сепарироваться

одним символом пробела), значения вещественной переменной типа double;

j. %mf – печать дампа памяти (байты значения, записанные в системе счисления с основанием 2, в порядке нахождения в памяти “слева направо”; строковые представления байтов должны сепарироваться одним символом пробела), значения вещественной переменной типа float.

Продемонстрируйте работу реализованных функций.

4. Реализуйте функции overfscanf и oversscanf, поведение которых схоже с поведением стандартных функций fscanf и sscanf соответственно, то есть эти функции имеют одинаковый прототип и логику работы, но в ваших функциях помимо стандартных флагов добавляются дополнительные флаги:
   a. %Ro – считывание из потока ввода целого числа типа int, записанного римскими цифрами;
   b. %Zr – считывание из потока ввода целого числа типа unsigned int, записанного в виде цекендорфова представления (коэффициенты 0 и 1 при числах Фибоначчи должны быть записаны от младшего к старшему слева направо с дополнительной единицей в конце записи, репрезентирующей окончание записи);
   c. %Cv считывание из потока ввода целого числа типа int, записанного в системе счисления с заданным основанием (при обработке флага первым параметром функции, “снимаемым” со стека, является адрес места в памяти типа int *, куда необходимо записать считанное значение, вторым – основание системы счисления (в которой записано число, находящееся в потоке ввода) в диапазоне [2..36] (при основании системы счисления, не входящем в диапазон, значение основания системы счисления устанавливается равным 10)); символы букв во входном строковом представлении целого числа должны быть записаны в нижнем регистре;
   d. %CV – аналогично флагу %Cv , при этом символы букв во входном строковом представлении целого числа должны быть записаны в верхнем регистре; Валидация данных, находящихся во входном потоке, не требуется.

Продемонстрируйте работу реализованных функций

5. Реализуйте программу, которая принимает на вход 2 строки(в интерактивном режиме) – пути к файлам ввода и вывода. Программа должна переносить текст из первого файла во второй файл. При этом производится разбиение каждой считанной строки исходного текста на строки по 80 символов, где 80-й – последняя буква некоторого слова (записанная без пробелов последовательность любых читаемых символов). Для этого между словами в строке равномерно добавляются пробелы. Если в считанном куске < 80 символов, он просто переписывается в новый файл. Первым символом новой строки должен быть читаемый символ
6. Экземпляр структуры типа Student содержит поля: id студента (целое неотрицательное число), имя (непустая строка только из букв латинского алфавита), фамилия (непустая строка только из букв латинского алфавита), группа (непустая строка) и оценки за 5 экзаменов (динамический массив

лементов типа unsigned char). Через аргументы командной строки программе на вход подаётся путь к файлу, содержащему записи о студентах. При старте программа считывает поданный файл в динамический массив структур типа Student. В программе должен быть реализован поиск всех студентов по:

- id;
- фамилии;
- имени;
- группе,

сортировка (для сортировки необходимо передавать компаратор для объектов структур) студента(-ов) по:

- id;
- фамилии;
- имени;
- группе.

Добавьте возможность вывода в трассировочный файл (путь к файлу передаётся как аргумент командной строки) данные найденного по id студента: ФИО, группу и среднюю оценку за экзамены. Также добавьте возможность вывести в трассировочный файл фамилии и имена студентов, чей средний балл за все экзамены выше среднего балла за все экзамены по всем считанным из файла студентам. Все вышеописанные опции должны быть выполнимы из контекста интерактивного диалога с пользователем. Для сортировки коллекции экземпляров структур используйте стандартную функцию qsort , своя реализация каких-либо алгоритмов сортировки не допускается.