Задача.

Рассмотрим ориентированный граф, подобный бинарному дереву без ссылок на родительскую вершину. Это означает, что в графе есть выделенная вершина, называемая корнем, каждая вершина имеет не более двух потомков, и каждое ребро графа определяется указателями left, right (возможно, нулевыми) на потомков и считается ориентированным от родителя к потомку. В отличие от бинарного дерева, здесь вершина может иметь несколько входящих ребер от разных "родительских" вершин. Такие вершины, у которых более одного входящего ребра, назовем особыми. а количество входящих ребер назовем рангом вершины.

Представление графа опирается на структуру Node. В базовом варианте структуры содержатся номер вершины (id) и указатели на потомков (left, right), определяющие ребра графа. Вам также даны функции, создающие граф по записям из файла, печатающие связи между вершинами графа. Функция печати реализует простейший аналог обхода сверху-вниз и поэтому поддеревья с корнями в особых вершинах печатаются несколько раз. Функции даны в исходных кодах. Для устранения возможных проблем с русскими кодировками комментарии в заготовках даны на английском языке.

К заготовке прилагается makefile со строгими ключами компиляции. Проверка вашей работы будет проводиться с использованием этого файла. Прежде чем вносить свои изменения, нужно скомпилировать и запустить полученную заготовку, например, командой make и последующим запуском ./a.out (или a.exe)

При реализации своих программ вы можете (и должны) добавлять новые поля и методы в структуру Node, можете модифицировать остальные процедуры. Однако надо понимать, что ответственность за работоспособность программы после внесенных изменений лежит полностью на вас. На эффективность программы особых требований не накладывается. Однако утечки памяти считаются ошибками.

К задаче прикладываются несколько тестовых файлов, определяющих различные графы. Следует проверить работу своей программы на всех данных файлах.

Задание. Требуется реализовать функцию, которая создает копию данного графа, нумерует вершины копии так старые номера с заданными приращением от старых номеров (т.е. idCopy = id + increment) и возвращает указатель на корневую вершину этой копии. Например, заголовок такой функции модет иметь вид

Node * Copy(Node * root, int increment);

Запрещается создание копии с использованием записи и чтения графа из файла.

Готовую программу надо протестировать на всех приложенных входных файлах и весь каталог с файлами вашей реализации и результатами тестов положить в один zip архив и отправить на почту своему преподавателю. Вывод результатов нужно выполнять в отдельный файл, чтобы проверяющий смог увидеть как ваша программа отработала на вашем компьютере. Так как ваша программа будет проверяться независимо от вас, то следует сделать выдачу интуитивно понятной, либо отдельно пояснить что в ней есть что. В заголовке письма нужно написать Nгруппы - фамилия - экзамен (скажем, 207-Садовничий-экзамен) имя архива также желательно назвать аналогичным образом.