**Лабоpатоpная pабота №2**

**Бинарные деревья**

Написать функцию формирования бинарного дерева, состоящего из целых чисел. Для представления дерева использовать динамические структуры данных. Количество элементов дерева, а также его вид задаются случайным образом. Произвести вывод элементов дерева тремя видами обхода. Используя информацию о выведенном дереве изобразить структуру одного из построенных деревьев в отчете. Выполнить по вариантам следующие задания:

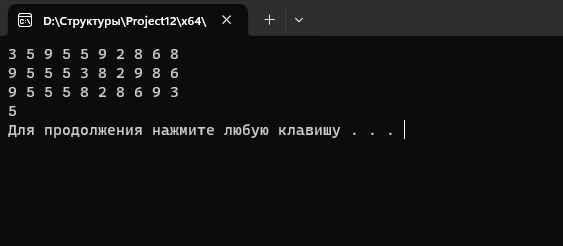
Вариант 12

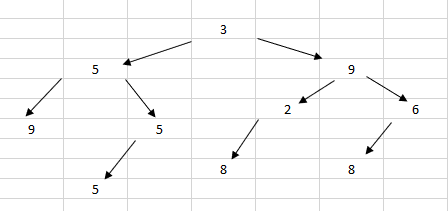
12. Написать рекурсивную функцию, которая находит элемент дерева, встречающийся чаще всего.

Код:

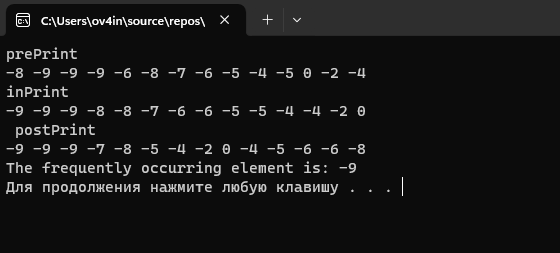
Тесты:

1)

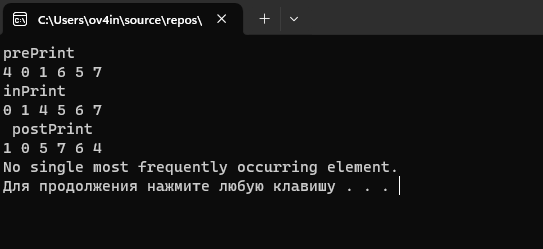




2)



3)



Алгоритм:

1. Определение структуры **Node** и функций для бинарного дерева:
   * Создание структуры **Node**, содержащей данные и указатели на левый и правый узлы.
   * Определение функции **add**, добавляющей новый узел в дерево.
2. Объявление корня бинарного дерева **root.**
3. Инициализация дерева с помощью функции **createTree:**
   * Создание случайного бинарного дерева с ограниченной вместимостью.
   * Генерация случайных значений и добавление их в дерево.
4. Вывод элементов дерева прямым обходом с помощью функции **prePrint:**
   * Рекурсивный обход дерева в прямом порядке (узел, левое поддерево, правое поддерево).
   * Вывод значений узлов в порядке их посещения.
5. Вывод элементов дерева симметричным обходом с помощью функции **inPrint:**
   * Рекурсивный симметричный обход дерева (левое поддерево, узел, правое поддерево).
   * Вывод значений узлов в порядке их посещения.
6. Вывод элементов дерева обратным обходом с помощью функции **postPrint:**
   * Рекурсивный обратный обход дерева (левое поддерево, правое поддерево, узел).
   * Вывод значений узлов в порядке их посещения.
7. Поиск наиболее часто встречающегося элемента в дереве с помощью функции **Search:**
   * Определение функции **findMaxFrequencyElement**, находящей узел с наибольшей частотой встречаемости.
   * Инициирование поиска наиболее часто встречающегося элемента в дереве.
   * Возврат значения наиболее часто встречающегося элемента или индикатора, если таких элементов несколько.

Алгоритм функции **findMaxFrequencyElement:**

1. **Начало с корня дерева:**

* + Функция начинает обход с корня дерева.
  + Рекурсивно переходит к левому поддереву узла.

1. **Проверка текущего узла:**
   * Для каждого узла проверяется его значение.
   * Если значение узла соответствует предыдущему обработанному элементу, увеличивается счетчик его частоты встречаемости.
   * Если значение отличается, обновляются текущий элемент и его частота.
2. **Обновление частоты и элемента с максимальной частотой:**
   * Если текущая частота больше максимальной, она становится новой максимальной, а текущий элемент - самым часто встречающимся элементом.
   * Если частоты равны, это может означать, что есть несколько элементов с максимальной частотой.
3. **Рекурсивный переход к правому поддереву:**
   * После обработки левого поддерева функция рекурсивно переходит к правому поддереву узла.
4. **Завершение обхода:**
   * После обхода всего дерева функция определяет наиболее часто встречающийся элемент и его частоту.
   * Если несколько элементов имеют одинаковую максимальную частоту, устанавливается флаг для обработки этого случая.