Отчёт по лабораторной работе № 24 по курсу Вычислительные системы

Студент группы М80-106Б Стрыгин Денис Дмитриевич, № по списку 22

Контакты www, e-mail, icq, skype _____

Работа выполнена: « 9 » апреля 2020 г.

Преподаватель: Ст. преп. каф.806 Дубинин А. В.

Входной контроль знаний с оценкой _____

Отчёт сдан « » апреля 2020 г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Деревья выражений

2. **Цель работы:** Составить программу выполнения заданных преобразований ариметических выражений с преминением деревьев

3. **Задание** (*вариант №*): Поделить многочлен от x на число k

4. **Оборудование(лабораторное)**:
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____Мб. Терминал _____адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:
Процессор _____ с ОП _____ Мб, НМД _____Мб. Монитор _____
Другие устройства _____

5. **Программное обеспечение(лабораторное):**
Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____.
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных на домашнем компьютере _____

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Считываем выражение через read_exp_res read_expression(queue *q) с помощью функции read_token и получаем очередь токенов. Затем, с помощью сортировочной станции Дейкстры (которая последовательно перекладывает токены из вход-) ной очереди сразу в выходную (для констант и переменных), в стэк операторы и скобки, причём ")" выталкивает из стека всё, что было до "(" и удаляет с ней, также оператор может направить в выхожную очередь несколько операторов (определяется ф-ей should+pop_out), а в конце все переносится в выходную очередь. Получаем обратную нотацию, из которой можно легко построить дерево выражений ф-ей built_tree. Затем выполнеятся обработка дерева выражений и его вывод в инфексной форме из дерева. Для того чтобы поделить многочлен (div_exp) нужно также написать функцию умножения многочлена на число k multi_exp по правилам деления дроби, т.е. её числитель отдельной ф-ией.

Для оператора деления - умножение на число k, для умножения - деление второго аргумента на число k, для сложения и вычита ния необходимо разделить оба аргумента, для степени - создание нового узла, который является копией узла возведения в степень и преобразование текущего узла под деление копии степени на число k, для унарного минуса необходимо проверить: если k<0, то убираем минус и делим аргументы бывшего минуса на число -k, для k>0 просто деление аргумента на k. Для константы деление на k. Для переменной - при k=1 ничего, при k=-1 - кмножение переменной на k, в остальных случаях добавление нового узла аналогично делению степени. При этом добавлении операнда деления/умножения вызывается функция check_const, которая добавляет унарный минус при отрицательном значении делителя/множителя, также она проверяет изменённое значение константы. Mulri_exp имеет тот же принцип обработки значения узла, но умножает. Также выполняется добавление унарного минуса для переменной, то есть выполнение деления/умножения происходит рекурсивно по правилам математики, при этом если нужно обработать только один аргумент из двух (при умножении), то обрабатывается первый

7. Сценарий выполнения работы [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

1.Написание token_tree, token_stack и treeptr_stack на основе созданных до этого int вариантов структур

2.Написание ф-ии для чтения выражения, преобразования в обратную польскую нотацию и дерево и их отладка

3. Отладка цикла чтения выражения

4.Написание ф-ии для выполнения задания

5.Тестирование программы на всевозможных выражениях

*Пункты 1-7 отчета составляются **строго до** начала лабораторной работы.*

Допущен к выполнению работы. Подпись преподавателя _____

8. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы _____

11. Выводы

Обработка выражений, записанных в виде дерева, вызывает основную сложность, однако, благодаря этой формы записи можно производить обработку наиболее эффективно.

Недочёты при выполнении задания могут быть устранены следующим образом:_____

Подпись студента_____