



Отчет по лабораторной работе № 7 по курсу Практикум на ЭВМ

Студент группы М8О-104Б-22 Жуков Дмитрий Алексеевич, № по списку 6

Контакты www, e-mail, icq, skype dmzhukov1@gmail.com

Работа выполнена: « 21 » октября 2022 г.

Преподаватель: асп. каф. 806 Потенко М.А.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Алгоритмическая модель Маркова

2. **Цель работы:** Написать рабочую программу для выполнения задания на эмуляторе алгоритмов Маркова

3. **Задание (вариант № 1):** Составить алгоритм, складывающий два троичных числа, разделенных знаком "+"

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 Windows _____, наименование Windows 10 _____ версия 2H21 _____
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 решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Алгоритм будет работать как машина Тьюринга, только вместо головки по тексту будет бегать какой-нибудь символ (не "0", "1", "2", "+"), который будет останавливаться сбоку от символа и перескакивать его, выполняя какую-нибудь операцию. Аналогом состояний МТ я буду использовать изменение символа головки на другой. Сам алгоритм будет уменьшать правое число на 1, а левое увеличивать на 1, после чего все, кроме левого числа сотрется. Таким образом, сначала перед числами возникает *, потом она, ничего не делая, доходит до крайнего правого положения. Там она превращается в точку и проходит второе число, уменьшая его на единицу, что легко делается заменой символов. После чего точка превращается в запятую, доходит к концу левого числа и превращается в '. Апостроф похожим образом пробегает справа налево по числу, но увеличивая его на единицу, после чего исчезает. В случае, если в начале второго числа мы будем видеть ноль, мы его будем стирать. Дальше опять слева появляется звездочка и так повторяется до тех пор, пока второе число не исчезнет полностью, т.е. пока мы не уменьшим его до нуля. Затем, в силу того, что условия на просто точку или плюс с точкой у нас нет, программа будет накидывать точки вправо от чисел, чем мы и воспользуемся. Такая ситуация возможна только после выполнения сложения, поэтому "+.." можно будет просто стереть, остановив программу.

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

Пока сложение не выполнено, слева от первого числа появляется *(19). Далее она пробегает оба числа(3-6) и оказывается справа от них. Меняем * на точку(7). Ее задача - уменьшить правое число на 1. Для этого она пробегает по всем нулям числа, заменяя их на двойки(8), а цифру, отличную от нуля, уменьшает на 1, обращаясь при этом в запятую(9-10). После этого запятая перебежит на конец первого числа и превратится в '(11-14). Апостроф увеличивает первое число на 1, после чего самоликвидируется(15-18). Стоит также обратить внимание на состояние (2) - оно убирает левый разряд второго числа, если тот уменьшается до нуля. При пошаговом ходе этого алгоритма заметим, что запись "X+Y" превращается в "Z+", после чего алгоритм начинает пытаться уменьшить второе число, которого нет. Алгоритм не знает, что делать с точкой, когда перед ней нет цифры, и дальше точки работа не идет - алгоритм начинает рождать звездочки, уводить их вправо и превращать в точки. Чтобы это прекратить, дадим условие завершения программы(1), которое гарантирует нам, что алгоритм точно закончится немного позже выполнения сложения, выдав ответом "Z".

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)

1. $+. \mapsto$

2. $+0 \rightarrow +$

3. $*0 \rightarrow 0*$

4. $*1 \rightarrow 1*$

5. $*2 \rightarrow 2*$

6. $*+ \rightarrow +*$

7. $* \rightarrow .$

8. $0. \rightarrow .2$

9. $1. \rightarrow 0,$

10. $2. \rightarrow 1,$

11. $2, \rightarrow ,2$

12. $1, \rightarrow ,1$

13. $0, \rightarrow ,0$

14. $+, \rightarrow '+$

15. $0' \rightarrow 1$

16. $1' \rightarrow 2$

17. $2' \rightarrow '0$

18. $' \rightarrow 1$

19. $\rightarrow *$

Тесты:

$0+1 \Rightarrow 1$

$1+0 \Rightarrow 1$

$0+0 \Rightarrow 0$

$2+2 \Rightarrow 11$

$1111+1010 \Rightarrow 2121$

$012+210 \Rightarrow 222$

$12+210 \Rightarrow 222$

$120+001 \Rightarrow 121$

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 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: Задача не затереть исходные данные не стояла

11. **Выводы:** В ходе данной работы мною были изучены алгоритмы Маркова. Несмотря на кажущуюся простоту, они могут выполнять широкий круг задач. Существует теорема, что все задачи, решаемые на МТ, решаемы и алгоритмами Маркова, и наоборот. Мною как раз был предложен, наверно, один из способов доказательства этой теоремы. Таким образом, я понял как не только осуществлять простые замены букв, но и проделывать простые арифметические операции, выполняемые МТ. Однако, стоит заметить, что Алгоритмы Маркова, наверно, еще более трудоемкие чем МТ, которая более трудоемкая, чем диаграммы Тьюринга, ведь в Алгоритмах Маркова изначально не предусмотрен функционал состояний, из-за чего его приходится изобретать с чистого листа, да и за порядком команд приходится строго следить.

12. **Недочёты при выполнении задания могут быть устранены следующим образом:** _____

Подпись студента _____