



Отчет по лабораторной работе № 07 по курсу Фундаментальная информатика

Студент группы М8О-104Б-22 Жарков Александр Алексеевич, № по списку 4

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

Работа выполнена: « 26 » 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:** Программирование в алгоритмической модели Маркова

2. **Цель работы:** Составить нормальный алгоритм Маркова, выполняющий преобразования над строкой

3. **Задание (вариант № 41):** Составить алгоритм вычисления троичного числа - троичного логарифма троичного числа

4. **Оборудование (лабораторное):**

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор AMD RYZEN 5 5600H с ОП 16384 Мб, ssd 524288 Мб. Монитор IPS 1920*1080

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 UNIX, наименование Ubuntu версия 20.04 LTS

интерпретатор команд bash версия 5.1.16

Система программирования _____ версия _____

Редактор текстов gedit версия 3.30.2

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)

Заменяем все цифры кроме незначащих нулей восклицательными знаками, и выведем их количество -1 (ведь троичный логарифм троичного числа равен количеству знаков в нём -1). Справа от ! добавим стоящие рядом символы @ и . (@ - конец восклицательных знаков, начало результата работы алгоритма. - конец результата работы алгоритма. Поодиночке будем переносить один ! слева в позицию между @ и и сразу прибавлять единицу к троичному числу-результату работы алгоритма. Таким образом количество восклицательных знаков слева будет уменьшаться на 1, а справа будет вестись подсчёт перенесённых восклицательных знаков в троичной СС (перенос единицы в следующий разряд будет выполняться при прибавлении единицы к двойке путём обнуления двойки и прибавления символа + слева от получившегося нуля, + будет прибавляться к ближайшему возможному разряду слева). После переноса всех ! вправо убираем символы @ и и завершаем работу алгоритма.

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

Заменяем все 1 и 2 на *, *0 на ** и 0* на * (выведем количество цифр в исходной строке, избавившись от незначащих нулей). Один раз заменим ** на !, далее будем заменять *! на !!, (добавив справа от всех ! символы @, стоящие рядом, которые понадобятся позднее), чтобы посчитать количество * - 1 (т.к. троичный логарифм троичного числа (округленный вниз) равен количеству цифр этого числа -1). Далее будем переносить по одному символу ! до положения !, будем прибавлять единицу к стоявшей слева от цифр. Если прибавляем единицу к двойке, то двойку заменяем на +0, прибавляем единицу к стоящему слева от + цифре. Если такой цифры нет, то есть в строке есть позиция @+. то заменяем + на 1. Если слева от + стоит 2, то также заменяем ее на +0 и повторяем операцию по переносу + влево. Таким образом мы считаем справа от @ и слева от количество символов ! сразу в троичной СС. Когда слева от @ заканчиваются !, мы заканчиваем подсчёт !, убираем символ @. убираем символ и заканчиваем работу алгоритма. Отдельно рассматриваем частные случаи: если в исходной строке 1 цифра (не считая незначащих нулей), она заменяется на *, * на 0 и работа алгоритма завершается. Если в исходной строке 2 цифры (не считая незначащих нулей), то они заменяются на **, ** на !, ! на 1 и работа алгоритма завершается.

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)

	Образец		Замена
1	!@#*	→	!!@#
2	0+	→	1
3	1+	→	2
4	2+	→	+0
5	!0#	→	1#
6	!0	→	0!
7	!1	→	1!
8	!2	→	2!
9	2!#	→	+0#
10	1!#	→	2#
11	0!#	→	1#
12	!0	→	1
13	!1#	→	2#
14	!2#	→	!0#
15	@!#	→	@1#
16	@+	→	@1
17	!@	→	@!
18	@	→	
19	#	→	.
20	1	→	*
21	2	→	*
22	*0	→	**
23	0*	→	*
24	!*	→	!!@#
25	**	→	!
26	*	→	0.
27	!	→	1.

Рабочая строка 1121212121

Рабочая строка 100

Рабочая строка 00000121212000210

Рабочая строка 102

Рабочая строка 00000001

Рабочая строка 0

Рабочая строка 0012

Рабочая строка 1

Рабочая строка 121221021

Рабочая строка 22

Рабочая строка 22110022110

Рабочая строка 101

Рабочая строка 120120120102102010210201021021212121100020021021020120

Рабочая строка 1222

Рабочая строка 01021020120102012010211012010000210201201021212120102000002110212021021012012121222201020120210012102020210201012021012

Рабочая строка 11100

Рабочая строка 0000000001111110000000

Рабочая строка 110

Рабочая строка 222000121

Рабочая строка 22

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: Использовано достаточно много специальных символов, в которых можно запутаться. Возможно, существует более простой алгоритм, решающий эту задачу.

11. **Выводы:** Я научился составлять нормальные алгоритмы Маркова и выполнил задачу. Работа с нормальными алгоритмами Маркова как и с Машинами Тьюринга помогает развить алгоритмическое мышление, полезное в программировании

Недочёты при выполнении задания могут быть устранены следующим образом: Перерешивание задачи, использование более простого алгоритма

Подпись студента _____