



Отчёт по лабораторной работе № 7 по курсу "Фундаментальная информатика"

Студент группы М8О-104Б-22, Полятыкин Никита, № по списку 13

Контакты www, e-mail, icq, skype polatykin58@gmail.com

Работа выполнена: « 8 » октября 20 22 г.

Преподаватель: Потенко М.А. каф.806

Входной контроль знаний с оценкой

Отчёт сдан « » 201 г., итоговая оценка

Подпись преподавателя

1. Тема: Программирование в алгоритмической модели Маркова

2. Цель работы: Составление программы в интерпретаторе алгоритмов Маркова, соответствующей данному варианту задания.

3. Задание (вариант № 9): Входное слово представляет собой два двоичных числа без знака, разделенных знаком ">". Составить алгоритм вычисления двоичного логического сдвига первого числа влево на числа разрядов, равное второму числу.

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 5500u 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 Linux, наименование Linux Ubuntu версия 22.04.1
интерпретатор команд Bash версия 5.1.16
Система программирования версия
Редактор текстов nano версия
Утилиты операционной системы Терминал

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)

Для решения данной задачи по сути надо составить два алгоритма, выполняющие следующие действия: вычитание из второго двоичного числа единицы и совершение логического сдвига влево на один знак второго двоичного числа. Программа должна быть написана так, чтобы сразу после выполнения первого алгоритма (вычитание единицы), начиналось выполнение второй (логический сдвиг на один знак), после чего всё начинается заново. Это значит, что алгоритм для счёта должен находится выше того, который отвечает за сдвиг. И тогда, когда второе число станет равно нулю (или вовсе будет отсутствовать), первое число будет сдвинуто на нужное количество разрядов и, соответственно, программа должна завершиться.

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

Вначале будет написана программа для вычитания из второго двоичного числа единицы. Для написания алгоритма придётся использовать маркеры. Вначале будет введён маркер "*", который будет пробегаться по второму числу слева направо ($>1 \rightarrow >*1$, $*00 \rightarrow 0*0$, $*01 \rightarrow 0*1$, $*10 \rightarrow 1*0$, $*11 \rightarrow 1*1$). Когда *-маркер дойдет до конца второго числа, то возможно два пути дальнейшего развития: 1) последней цифрой оказывается "1", тогда алгоритм производит замену $*1 \rightarrow 0!$ (т.е. происходит вычитание единицы из последнего числа), введён маркер "!", который бежит через всё второе число справа налево ($0! \rightarrow !0$, $0!0 \rightarrow !00$, $0!1 \rightarrow !01$, $1!0 \rightarrow !10$, $1!1 \rightarrow !11$, $1! \rightarrow !1$), а когда !-маркер дойдёт до знака ">", появится маркер "/" ($>! \rightarrow >/$), с которого и начинается выполнение второй части алгоритма (сдвиг); 2) последняя цифра - "0" - здесь же происходит чуть больше действий: вводится маркер "@", он пробегает число справа налево ($00 \rightarrow 0,0$) вплоть до того, пока не обнаружит единицу и алгоритм производит замену $1,0 \rightarrow 0@1$, тогда же @-маркер бежит направо, заменяя все нули на единицы ($1@0 \rightarrow 11@$), т.е. произошло вычитание единицы; после этого, заменив все нули на единицы, @-маркер заменяется на ! и повторяются действия, указанные выше. Также надо учесть, что если первой цифрой правого числа становится ноль, то программа должна его удалить ($>*0 \rightarrow >*$). Теперь алгоритм начинает выполнять вторую часть - двоичный логический сдвиг влево первого числа. Для логического сдвига влево надо, чтобы справа от числа появился ноль, а левая цифра была откинута. Маркер "/", оказавшись справа от первого числа, перепрыгивает первые два числа справа, добавляя к ним ноль ($00/ \rightarrow /000$, $01/ \rightarrow /010$, $10/ \rightarrow /100$, $11/ \rightarrow /110$), после чего пробегается до левого конца числа ($0/0 \rightarrow /00$, $0/1 \rightarrow /01$, $1/0 \rightarrow /10$, $1/1 \rightarrow /11$). Когда /-маркер доходит до левого конца, крайнее число удаляется вместе с маркером ($/0 \rightarrow$, $/1 \rightarrow$). Следовательно, строка находится в таком же состоянии, как и в начале работы программы (без каких-либо маркеров), и алгоритм начинает работать снова с первой части (т.е. вычитать единицу во втором числе). Программа должна успешно завершиться, когда она обнаружит сочетание ">*" и ничего больше - это будет означать, что второе число было сёрто, а первое - сдвинуто на соответствующее количество разрядов. Замена $>* \Rightarrow$ завершает работу программы. В качестве тестов будут использованы два однозначных числа - 0 и 1 - разделенные знаком ">" с числом "111" и "1110" (результат в обоих случаях - "0"), также "111>1000" (результат - "000") и "111101110>11" (результат - "101110000").

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

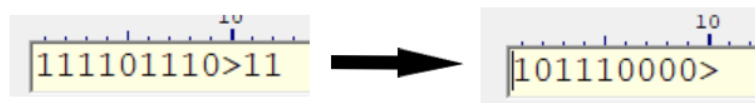
Допущен к выполнению работы. Подпись преподавателя _____

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

	Образец	→	Замена
1	010	→	110
2	100	→	110
3	10	→	11
4	01	→	11
5	010	→	110
6	>*0	→	>*0
7	*00	→	0*0
8	*01	→	0*1
9	*10	→	1*0
10	*11	→	1*1
11	*1	→	0!
12	*0	→	,0
13	0,0	→	,00
14	1,0	→	001
15		→	
16	> ,	→	.
17	>*	→	.
18	0!	→	!0
19	0!0	→	!00
20	0!1	→	!01
21	1!0	→	!10
22	1!1	→	!11
23	1!	→	!1
24		→	

	Образец	→	Замена
16	> ,	→	.
17	>*	→	.
18	0!	→	!0
19	0!0	→	!00
20	0!1	→	!01
21	1!0	→	!10
22	1!1	→	!11
23	1!	→	!1
24		→	
25		→	
26		→	
27	>!	→	/>
28		→	
29		→	
30		→	
31	/0>	→	0>*
32	0/0	→	/00
33	0/1	→	/01
34	1/0	→	/10
35	1/1	→	/11
36	/0	→	
37	/1	→	
38	00/	→	/000
39	01/	→	/010

	Образец	→	Замена
26		→	
27	>!	→	/>
28		→	
29		→	
30		→	
31	/0>	→	0>*
32	0/0	→	/00
33	0/1	→	/01
34	1/0	→	/10
35	1/1	→	/11
36	/0	→	
37	/1	→	
38	00/	→	/000
39	01/	→	/010
40	10/	→	/100
41	11/	→	/110
42	>0	→	>
43	0/>	→	0>
44	1/>	→	0>
45	>1	→	>*1



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

В программе для выполнения поставленной задачи используется пять маркеров (* ! , @ /). Предполагаю, работа могла быть выполнена с использованием меньшего количества маркеров.

11. Выводы

Я составил алгоритм в интерпретаторе алгоритмов Маркова, выполняющий поставленную задачу. Нормальный алгоритм Маркова, наряду с машиной Тьюринга, является одним из стандартных способов определения понятия алгоритм, и поэтому умение составлять НАМ поможет в дальнейшей работе по написанию различных программ.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента

