

Отчет по лабораторной работе №5

по курсу _____

Студент группы: М8О-101Б-22, Шляхтуров Александр Викторович, № по списку: 26, Контакты

shliakhturov@gmail.com Работа выполнена: «1» октября 2022г.

Преподаватель: каф. 806 Крылов Сергей Сергеевич,

Входной контроль знаний с оценкой _____

Отчет сдан « 3 » 09 2022 г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Программирование машин Тьюринга

2. **Цель работы:** Составить программу машины Тьюринга в четвёрках, выполняющую заданное действие над словами, записанными на ленте.

2. **Задание (вариант №26):** Проверить палиндромию двоичного числа

3. Оборудование

Оборудование ПЭВМ студента, если использовалось:

Процессор AMD Ryzen 5 5600, 6 ядер с ОП 16 гб, ТТН 512 гб

6. Идея, метод

Программа работает следующим образом:

Головка передвигается в начало слова, заменяя первый символ на пробел. В зависимости от замененного символа (1/0) запускается определенная цепочка состояний. Головка передвигается в конец исходного слова, пропуская один пробел, и возвращает «украденный» из начала символ. Далее, следуя той же цепочке состояний, головка проходит возвращенный символ, первый пробел, разделяющий исходное слово и второе слово-копию, и все исходное слово до его начала. Цепочка состояний заканчивается командой, которая возвращает «украденный символ» в начало слова, тем самым, восстанавливая его до первоначального состояния. Цепочка состояний завершается, переходя к командам, передвигающим головку на одну позицию вправо – на следующий символ исходного слова. Далее также в зависимости от символа запускается цепочка состояний, которая снова заменяет его на пробел, передвигает головку до первого пробела, проходит начало второго слова, и ставит следующим символом во втором слове. Затем головка передвигается до места, откуда был «украден» символ первого слова и восстанавливает, заменяя пробел на первоначальный символ. Таким образом слово копируется посимвольно справа от исходного.

Следующий шаг – проверка слова на палиндромию. Головка перемещается в начало второго слова – точную копию первого, заменяя первый символ на пробел – крадет его, подобно цепочке состояний из первого абзаца. Далее запускается цепочка состояний, которая перемещает головку в конец второго слова. Цепочка состояний такова, что если «украденный» символ совпадает с последним символом, то последний заменяется на пробел, а головка возвращается в начало второго слова. В случае, если первый символ второго слова не совпадает с последним – запускается завершающая цепочка состояний, функция которой – стереть второе слово и вывести ноль рядом с исходным словом через пробел. В случае несовпадения крайних символов, слово очевидно не является палиндромом, индикатором чему является ноль.

По такой схеме, головка будет заменять на пробелы крайние символы слова-копии, уменьшая его до тех пор, пока они

- 1) либо не совпадут
- 2) либо пока не останется единственный символ (в случае нечетного количества символов)
- 3) либо пока не останется символов вовсе.

Второй и третий пункт свидетельствуют о палиндромии слова. В этом случае головка удаляет слово-копию и выводит единицу через пробел от исходного слова. Единица свидетельствует о том, что исходное слово - палиндром.

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

Входные данные	Выходные данные	Описание тестируемого случая
0	0 1	Число 0 является палиндромом
1	1 1	Число 1 является палиндромом
10	10 0	Число 10 не является палиндромом

1001001001001	1001001001001 1	Число 1001001001001 является палиндромом
---------------	-----------------	--

8. Распечатка протокола

КОМАНДЫ ДЛЯ МАШИНЫ ТЬЮРИНГА С ОБОЗНАЧЕНИЯМИ СОСТОЯНИЙ	КОМАНДЫ ДЛЯ МАШИНЫ ТЬЮРИНГА С ПОРЯДКОВОЙ НУМЕРАЦИЕЙ СОСТОЯНИЙ
00, <,START	00, <,01
START,1,<,START // Прохождение числа справа налево	01,1,<,01
START,0,<,START	01,0,<,01
START, >,eat	01, >,02
	02,0, ,03
eat,0, ,ate_0	02,1, ,04
eat,1, ,ate_1	02, >,15
eat, >,PROGRAM_START	03,1,>,03
	03,0,>,03
	03, >,05
// КОПИРОВАНИЕ ЧИСЛА	04,1,>,04
ate_0,1,>,ate_0 // Съедаем первое число (0) и вставляем через пробел	04,0,>,04
ate_0,0,>,ate_0	04, >,08
ate_0, >,place_0	05,0,>,05
	05,1,>,05
	05, >,06
ate_1,1,>,ate_1 // Съедаем первое число (1) и вставляем через пробел	06,1,>,06
ate_1,0,>,ate_1	06,0,>,06
ate_1, >,place_1	06, ,07
	07,1,<,07
place_0,0,>,place_0	07,0,<,07
place_0,1,>,place_0	07, <,11
place_0, >,place_0_end	08,0,>,08
	08,1,>,08
place_0_end,1,>,place_0_end // Едем назад	08, >,09
place_0_end,0,>,place_0_end	09,1,>,09
place_0_end, ,0,back_0_pre	09,0,>,09
	09, ,1,10
back_0_pre,1,<,back_0_pre	10,1,<,10
back_0_pre,0,<,back_0_pre	10,0,<,10
back_0_pre, <,back_0	10, <,13
	11,0,<,11
place_1,0,>,place_1	11,1,<,11
place_1,1,>,place_1	11, ,0,12
place_1, >,place_1_end	12,0,>,02
	13,0,<,13
place_1_end,1,>,place_1_end	13,1,<,13
place_1_end,0,>,place_1_end	13, ,1,14
place_1_end, ,1,back_1_pre	14,1,>,02
	15,1, ,16
back_1_pre,1,<,back_1_pre	15,0, ,17
back_1_pre,0,<,back_1_pre	15, >,26
back_1_pre, <,back_1	16, >,18
	17, >,19
back_0,0,<,back_0	18,1,>,18
back_0,1,<,back_0	18,0,>,18
back_0, ,0,eat_next_pre	18, <,21
eat_next_pre,0,>,eat	19,1,>,19
	19,0,>,19
back_1,0,<,back_1	19, <,20
back_1,1,<,back_1	20,0, ,23
back_1, ,1,eat_next_pre1	20,1, ,30
eat_next_pre1,1,>,eat	20, <,26
	21, <,26
// ЧИСЛО СКОПИРОВАНО, ТЕПЕРЬ ПРОВЕРЯЕМ ЕГО НА ПОЛИНДРОМ	21,1, ,22
PROGRAM_START,1, ,del_1_left_pre	21,0, ,30
PROGRAM_START,0, ,del_0_left_pre	22, <,24
	23, <,25
PROGRAM_START, >,GOOD_delete_n_exit	24,1,<,24
	24,0,<,24
del_1_left_pre, >,del_1_left	24, >,15
	25,1,<,25
del_0_left_pre, >,del_0_left	25,0,<,25
	25, >,15
del_1_left,1,>,del_1_left // Запоминаем число слева и едем направо	26, <,26
	26,1,>,27
	26,0,>,27
	27, >,28
	28, ,1,28
	28,1,>,29
	29, #,29
	30, <,31
	31,1, ,32

del_1_left,0,>,del_1_left del_1_left,<,<,del_1_right del_0_left,1,>,del_0_left del_0_left,0,>,del_0_left del_0_left,<,<,del_0_right del_0_right,0,>,go_back_0_pre // Удаляем число справа и, если оно не совпало с тем, что мы удалили, del_0_right,1,>,BAD_delete_n_exit // удаляем остатки скопированного числа и выходим со значением 0 // иначе возвращаемся направо и цикл начинается заново del_0_right,<,<,GOOD_delete_n_exit del_1_right,<,<,GOOD_delete_n_exit del_1_right,1,>,go_back_1_pre del_1_right,0,>,BAD_delete_n_exit go_back_1_pre,<,<,go_back_1 go_back_0_pre,<,<,go_back_0 go_back_1,1,>,go_back_1 go_back_1,0,>,go_back_1 go_back_1,>,PROGRAM_START go_back_0,1,>,go_back_0 go_back_0,0,>,go_back_0 go_back_0,>,PROGRAM_START GOOD_delete_n_exit,<,<,GOOD_delete_n_exit GOOD_delete_n_exit,1,>,GOOD_END01 GOOD_delete_n_exit,0,>,GOOD_END01 GOOD_END01,>,GOOD_END GOOD_END,>,1,GOOD_END GOOD_END,1,>,GOOD_END_END GOOD_END_END,>,>,GOOD_END_END BAD_delete_n_exit,<,<,BAD_delete BAD_delete,1,>,BAD_delete_2 BAD_delete,0,>,BAD_delete_2 BAD_delete_2,<,<,BAD_delete BAD_delete_space,<,<,BAD_delete BAD_delete,<,<,BAD_END01 BAD_delete_n_exit,1,>,BAD_END01 BAD_delete_n_exit,0,>,BAD_END01 BAD_END01,<,<,BAD_END01 BAD_END01,0,>,babus1 BAD_END01,1,>,babus1 babus1,>,BAD_END BAD_END,>,0,BAD_END BAD_END,0,>,BAD_END_END BAD_END_END,>,>,BAD_END_END	31,0,>,32 32,<,<,31 33,<,<,31 31,<,<,34 30,1,>,34 30,0,>,34 34,<,<,34 34,0,>,35 34,1,>,35 35,>,36 36,>,36 36,0,>,37 37,>,>,37
--	---

...

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
0	дом	01.10.2022	04:20	Случайно закрыл браузер и стер код	Переписал код заново	_____

10. Замечания автора

11. Выводы

Работу считаю актуальной для студентов – новичков в программировании, коим я и являюсь. Могу образно сказать, что стал «умнее» во время ее осуществления, поскольку пришлось серьезно озадачиваться, чтобы придумать ход решения задачи. Программирование машины Тьюринга – хорошая тренировка для памяти и абстрактного мышления, которая пойдет всем на пользу.

Подпись студента _____