

# Отчет по лабораторной работе № 5 по курсу “Фундаментальная информатика”

Студент группы М8О-109Б-22 Нгуен Ньы Хоанг Ань, № по списку 12

Контакты vk, telegram @tng00

Работа выполнена: «27» сентября 2022 г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » 2022 г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Программирование машин Тьюринга
2. **Цель работы:** Изучение и освоение машины Тьюринга, создание программы на эмуляторе машины Тьюринга в четвёрках \_\_\_\_\_.
3. **Задание (вариант 7):** Генерация двух чисел из разрядов двоичного числа, находящихся на чётных и нечётных позициях.
4. **Оборудование (студента):**  
Процессор *Intel(R) Core(TM) i3-7100U CPU @ 2.40GHz* с ОП 5,88 Гб, НМД 1024 Гб. Монитор 1920x1080
5. **Программное обеспечение (студента):**  
Операционная система семейства: *linux*, наименование: *ubuntu*, версия *22.04.1 LTS*  
Интерпретатор команд: *bash* версия *5.1.16*.  
Система программирования -- версия --, редактор текстов *emacs* версия *28.2*  
Утилиты операционной системы --  
Прикладные системы и программы --  
Местонахождение и имена файлов программ и данных на домашнем компьютере */home/tng00*

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

Поочерёдно перемещать вправо от исходного числа сначала нечётные разряды, затем чётные. Пробел использовать как маркер того, что с некой позиции взяли цифру. Для простоты восприятия команды будут содержать обозначения: буквы *o* – odd (нечётный) и *e* – even (чётный), цифра, стоящая после буквы, является задействованной в данный момент.

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

План работы:

1. Перемещаемся к пробелу перед исходным числом, затем к первой цифре
2. Берём первый нечётный разряд, на его место ставим пробел (как маркер)
3. Перемещаем цифру вправо от исходного числа, отделив её пробелом
4. Возвращаем цифру на позицию в исходном числе (место пробела)
5. Переходим к следующим нечётным разрядам и проделываем то же самое
6. Как только нечётные разряды закончатся, окажемся в промежутке между исходным и первым генерированным числом
7. Перемещаемся к пробелу перед числом, затем ко второй цифре
8. Повторяем действия 2-5 для чётных разрядов с одним отличием: цифры перемещаем вправо от первого генерированного числа
9. Как только чётные разряды закончатся, окажемся в промежутке между исходным и первым генерированным числом
10. Перемещаемся в позицию за вторым генерированным числом. Работа машины Тьюринга закончена.

Тесты и результат:

1. 0 - 0
2. 10 - 1 0
3. 001 - 01 0
4. 101010101 – 11111 0000

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

```
00, ,<,00- // переход к началу числа
00-,0,<,00-
00-,1,<,00-
00-, ,>,01
```

```
// Работа с нечётными разрядами
```

```
01,0, ,02o0 // замена на пробел
01,1, ,02o1
01o, , ,08o0
```

```
02o0, ,>,03o0 // переход в крайнюю правую позицию заданного числа
02o1, ,>,03o1
03o0,0,>,03o0
03o0,1,>,03o0
03o0, ,>,04o0
03o1,0,>,03o1
03o1,1,>,03o1
03o1, ,>,04o1
```

```
04o0, ,0,05o0 // копирование цифры
04o0,0,>,04o0
04o0,1,>,04o0
04o1, ,1,05o1
04o1,0,>,04o1
04o1,1,>,04o1
```

```
05o0, ,<,05o0- // переход к исходной позиции взятой цифры
05o0,0,<,05o0
05o0,1,<,05o0
05o1, ,<,05o1-
05o1,0,<,05o1
05o1,1,<,05o1
```

```
05o0-, ,0,06o0 // возврат взятой цифры
05o0-,0,<,05o0-
05o0-,1,<,05o0-
05o1-, ,1,06o1
05o1-,0,<,05o1-
05o1-,1,<,05o1-
```

```
06o0-, , ,07o0 // переход к работе со следующей цифрой
06o0-,0,>,07o0
06o0-,1,>,07o0
06o0,0,>,06o0-
06o1-, , ,07o1
06o1-,0,>,07o1
06o1-,1,>,07o1
06o1,1,>,06o1-
```

```
08o0, ,>,08o0- // переход от работы с нечётными к чётным цифрам
08o0-,0,>,01e
08o0-,1,>,01e
08o0,0,<,08o0
08o0,1,<,08o0
08o1, ,>,08o0-
08o1,0,<,08o1
08o1,1,<,08o1
```

```
07o0, ,<,08o0 // первая чётная цифра
07o0,0,0,01
07o0,1,1,01
07o1, ,<,08o1
07o1,0,0,01
```

0701,1,1,01

// Работа с чётными разрядами

01e0, ,02e0 // замена на пробел  
01e1, ,02e1  
01e, , ,07e0-

02e0, ,>,03e0 // переход в крайнюю правую позицию заданного числа  
02e1, ,>,03e1  
03e0,0,>,03e0  
03e0,1,>,03e0  
03e0, ,>,04e0-  
03e1,0,>,03e1  
03e1,1,>,03e1  
03e1, ,>,04e1-

04e0-,0,>,04e0- // переход в крайнюю правую позицию первого результата  
04e0-,1,>,04e0-  
04e0-, ,>,04e0  
04e1-,0,>,04e1-  
04e1-,1,>,04e1-  
04e1-, ,>,04e1

04e0, ,0,05e0 // копирование цифры  
04e0,0,>,04e0  
04e0,1,>,04e0  
04e1, ,1,05e1  
04e1,0,>,04e1  
04e1,1,>,04e1

05e0, ,<,05e0-- // обратный путь через первый результат  
05e0,0,<,05e0  
05e0,1,<,05e0  
05e1, ,<,05e1--  
05e1,0,<,05e1  
05e1,1,<,05e1

05e0--, ,<,05e0- // переход к исходной позиции взятой цифры  
05e0--,0,<,05e0--  
05e0--,1,<,05e0--  
05e1--, ,<,05e1-  
05e1--,0,<,05e1--  
05e1--,1,<,05e1--

05e0-, ,0,06e0 // возврат взятой цифры  
05e0-,0,<,05e0-  
05e0-,1,<,05e0-  
05e1-, ,1,06e1  
05e1-,0,<,05e1-  
05e1-,1,<,05e1-

06e0-, , ,07e0 // переход к работе со следующей цифрой  
06e0-,0,>,07e0  
06e0-,1,>,07e0  
06e0,0,>,06e0-  
06e1-, , ,07e1  
06e1-,0,>,07e1  
06e1-,1,>,07e1  
06e1,1,>,06e1-

// Конец программы

07e0, ,>,07e0-  
07e0---, , ,07e0---  
07e0--, , ,07e0---

07e0--0,>,07e0--  
07e0--1,>,07e0--  
07e0-,>,07e0--  
07e0-0,>,07e0-  
07e0-1,>,07e0-  
07e0,0,0,01e  
07e0,1,1,01e  
07e1,>,07e1-  
07e1---, ,07e1---  
07e1--, ,07e1---  
07e1--,0,>,07e1--  
07e1--,1,>,07e1--  
07e1-,>,07e1--  
07e1-0,>,07e1-  
07e1-1,>,07e1-  
07e1,0,0,01e  
07e1,1,1,01e

**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**

Лабораторная работа помогла понять, как работает машина Тьюринга. Было интересно писать программу на языке низкого уровня, наблюдать, как привычные алгоритмы реализуются с помощью последовательности примитивных действий.

Подпись студента 