

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

Студент группы М80-101Б-22, Бычков Артур Сергеевич, № по списку 2

Контакты email: bychkovarthur@gmail.com

Работа выполнена: «1» октября 2022г.

Преподаватель: каф. 806 Крылов Сергей Сергеевич

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202__ 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.

3. **Задание** (вариант № 2*): Вычисление поразрядной дизъюнкции двух двоичных чисел.

4. **Оборудование:**

Оборудование ПЭВМ студента, если использовалось:

Процессор **AMD Ryzen 5 5500U 2.10 GHz, 6 ядер** с ОП **8192 Мб**, ТТН **512000 Мб**. Мониторы **Lenovo**.

5. **Программное обеспечение:**

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства **Linux**, наименование **Ubuntu** версия **20.04.5**, интерпретатор команд bash версия 5.0.17(1).

Система программирования Clion версия 2021.1.3

Редактор текстов nano версия 6.2

Утилиты операционной системы WinRar, Microsoft Word.

Прикладные системы и программы Ubuntu wsl, Clion, Google Chrome

Местонахождение и имена файлов программ и данных на домашнем компьютере /home/artur

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

Моя идея такова:

- 1) Я считываю по 1 самой правой цифре из каждого числа, если это комбинации 01, 10, 11, то я печатаю справа 1, иначе, если это 00, то я печатаю 0
- 2) Если исходные числа имеют разную длину, я их считываю как в пункте 1), но когда дохожу до того момента, когда одно число закончилось, а второе еще нет, то я попросту копирую то, что осталось от более длинного числа.
- 3) В итоге у меня выходит ответ, однако он перевернут (вместо 100 у меня будет написано 001). Я просто беру и переворачиваю ответ.

Как я это реализовал:

- 1) Когда я считываю числа, я их запоминаю и меняю на пробел, после ставлю в ответе 0 или 1 в зависимости от того, какие числа были изначально.
- 2) Я начинаю возвращаться и, когда встречаю пробел не на месте разделения чисел (внутри одного из чисел), я ставлю то число, которое здесь было на место, а следующее запоминаю и меняю его на пробел.
- 3) Пункт 2) продолжается до тех пор, пока я не встречу 2 пробела подряд, это значит, что одно из чисел закончилось. Если закончилось только одно из чисел, то я алгоритмом из предыдущих пунктов копирую оставшееся число, а после иду в правый конец, если оба числа закончились я сразу иду в самый правый конец.
- 4) Переворот числа: Сначала я запоминаю крайнее правое и крайнее левое числа, и вместо них ставлю пробел, после я снова прохожусь по числу если нахожу пробел, который стоит внутри числа, то я на его места ставлю цифры, которые я прочитал на прошлом ходу, только наоборот, т.е. в левую позицию я ставлю правое число, а в правую позицию левое число. После того, как я поставил числа я считываю следующее (оно расположено ближе к центру числа, т.е. для правой позиции это число расположено на 1 левее, а для левой на 1 правее), на его место так же ставлю пробел.
- 5) После каждого запоминания и постановки пары цифр, я делаю проверку на то, является ли это действие последним: а) Если внутри числа расположены 2 пробела подряд, например 101 110, то я просто ставлю числа левое на правое место, а правое на левое и завершаю программу, б) Если внутри числа два пробела расположены через 1 цифру, например, 100 1 011, то я так же ставлю левую цифру на правое место, правую на левое и завершаю программу.

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

<i>Входные данные</i>	<i>Выходные данные</i>	<i>Описание тестируемого случая</i>
0 0	0 0 0	Все случаи с однозначными числами
0 1	0 1 1	
1 0	1 0 1	
1 1	1 1 1	
00 00	00 00 00	Все случаи с двузначными числами
00 01	00 01 01	
00 10	00 10 10	
00 11	00 11 11	
01 00	01 00 01	
01 01	01 01 01	
01 10	01 10 11	
01 11	01 11 11	
10 00	10 00 10	
10 01	10 01 11	
10 10	10 10 10	
10 11	10 11 11	
11 00	11 00 11	
11 01	11 01 11	
11 10	11 10 11	
11 11	11 11 11	
01010101 10101010	01010101 10101010 11111111	Проверка на то, не появятся ли лишние цифры (должны быть только 1 в обоих случаях)
10101010 01010101	10101010 01010101 11111111	
0000000000 0000000000	0000000000 0000000000 0000000000	Одинаковые длинные числа
1111111111 1111111111	1111111111 1111111111 1111111111	
00000 1	00000 1 00001	Одно однозначное число – другое
1 00000	1 00000 00001	

11111 0	11111 0 11111	пятизначное, состоящее из другой цифры
0 11111	0 11111 11111	
1001 1010011	1001 1010011 1011011	Случайные тесты
10111101 10100010	10111101 10100010 10111111	
1 1001001010	1 1001001010 1001001011	
1010101010010 1	1010101010010 1 1010101010011	
1010010 10110	1010010 10110 1010110	
11111111 10101001010	11111111 10101001010 10111111111	
010000000 1	010000000 1 010000001	
1010	101	Дано всего одно из двух чисел
	Выход за границу ленты	
001110	00111	
	Выход за границу ленты	
		Чисел нет
	Выход за границу ленты	

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

```
0, <,sec_num_?_?
sec_num_?_?,1, ,?_1
sec_num_?_?,0, ,?_0
?_1, <,on_sec_?_1
on_sec_?_1,0,<,on_sec_?_1
on_sec_?_1,1,<,on_sec_?_1
on_sec_?_1, <,fst_num_?_1
?_0, <,on_sec_?_0
on_sec_?_0,0,<,on_sec_?_0
on_sec_?_0,1,<,on_sec_?_0
on_sec_?_0, <,fst_num_?_0
// Первое считывание двух цифр
fst_num_?_1,0, ,go_to_place01_fst_sp0
go_to_place01_fst_sp0, >,go_to_place01_fst_sp1
fst_num_?_1,1, ,go_to_place11_fst_sp0
go_to_place11_fst_sp0, >,go_to_place11_fst_sp1
fst_num_?_0,1, ,go_to_place10_fst_sp0
go_to_place10_fst_sp0, >,go_to_place10_fst_sp1
fst_num_?_0,0, ,go_to_place00_fst_sp0
go_to_place00_fst_sp0, >,go_to_place00_fst_sp1
// Постановка результата поразрядной дизъюнкции для 00
go_to_place00_fst_sp1,1,>,go_to_place00_fst_sp1
go_to_place00_fst_sp1,0,>,go_to_place00_fst_sp1
go_to_place00_fst_sp1, >,go_to_place00_sec_sp0
go_to_place00_sec_sp0,1,>,go_to_place00_sec_sp0
go_to_place00_sec_sp0,0,>,go_to_place00_sec_sp0
go_to_place00_sec_sp0, >,go_to_place00_sec_sp1
go_to_place00_sec_sp1,1,>,go_to_place00_sec_sp1
go_to_place00_sec_sp1,0,>,go_to_place00_sec_sp1
go_to_place00_sec_sp1, >,placing00
placing00, ,0,go_to_start00
placing00,1,>,placing00
placing00,0,>,placing00
// Возвращение к началу для 00
go_to_start00,1,<,go_to_start00
go_to_start00,0,<,go_to_start00
go_to_start00, =,start00
// Постановка результата поразрядной дизъюнкции для 01
go_to_place01_fst_sp1,1,>,go_to_place01_fst_sp1
go_to_place01_fst_sp1,0,>,go_to_place01_fst_sp1
go_to_place01_fst_sp1, >,go_to_place01_sec_sp0
go_to_place01_sec_sp0,1,>,go_to_place01_sec_sp0
go_to_place01_sec_sp0,0,>,go_to_place01_sec_sp0
go_to_place01_sec_sp0, >,go_to_place01_sec_sp1
go_to_place01_sec_sp1,1,>,go_to_place01_sec_sp1
go_to_place01_sec_sp1,0,>,go_to_place01_sec_sp1
go_to_place01_sec_sp1, >,placing01
placing01, ,1,go_to_start01
placing01,1,>,placing01
placing01,0,>,placing01
// Возвращение к началу для 01
go_to_start01,1,<,go_to_start01
go_to_start01,0,<,go_to_start01
go_to_start01, =,start01
// Постановка результата поразрядной дизъюнкции для 10
go_to_place10_fst_sp1,1,>,go_to_place10_fst_sp1
go_to_place10_fst_sp1,0,>,go_to_place10_fst_sp1
go_to_place10_fst_sp1, >,go_to_place10_sec_sp0
go_to_place10_sec_sp0,1,>,go_to_place10_sec_sp0
go_to_place10_sec_sp0,0,>,go_to_place10_sec_sp0
go_to_place10_sec_sp0, >,go_to_place10_sec_sp1
go_to_place10_sec_sp1,1,>,go_to_place10_sec_sp1
go_to_place10_sec_sp1,0,>,go_to_place10_sec_sp1
go_to_place10_sec_sp1, >,placing10
placing10, ,1,go_to_start10
placing10,1,>,placing10
```

```

placing10,0,>,placing10
// Возвращение к началу для 10
go_to_start10,1,<,go_to_start10
go_to_start10,0,<,go_to_start10
go_to_start10,=,start10
// Постановка результата поразрядной дизъюнкции для 11
go_to_place11_fst_sp1,1,>,go_to_place11_fst_sp1
go_to_place11_fst_sp1,0,>,go_to_place11_fst_sp1
go_to_place11_fst_sp1,=,go_to_place11_sec_sp0
go_to_place11_sec_sp0,1,>,go_to_place11_sec_sp0
go_to_place11_sec_sp0,0,>,go_to_place11_sec_sp0
go_to_place11_sec_sp0,=,go_to_place11_sec_sp1
go_to_place11_sec_sp1,1,>,go_to_place11_sec_sp1
go_to_place11_sec_sp1,0,>,go_to_place11_sec_sp1
go_to_place11_sec_sp1,=,placing11
placing11,=,1,go_to_start11
placing11,1,>,placing11
placing11,0,>,placing11
// Возвращение к началу для 11
go_to_start11,1,<,go_to_start11
go_to_start11,0,<,go_to_start11
go_to_start11,=,start11
// Последующее возвращение исходных значений + прочтение новых для 00
start00,=,<,sec_placing_00_reading_**
sec_placing_00_reading_**,1,<,sec_placing_00_reading_**
sec_placing_00_reading_**,0,<,sec_placing_00_reading_**
// Ставим 0 во втором числе
sec_placing_00_reading_**,0,sec_placing_00_reading_**_sp1
sec_placing_00_reading_**_sp1,0,<,sec_placing_00_reading_**_sp1_0
sec_placing_00_reading_**_sp1,1,<,sec_placing_00_reading_**_sp1_0
// Переход к первому числу
sec_placing_00_reading_**_sp1_0,0,sec_placing_00_reading_0_sp1
sec_placing_00_reading_0_sp1,=,<,sec_placing_00_reading_0_sp1_1
sec_placing_00_reading_0_sp1_1,1,<,sec_placing_00_reading_0_sp1_1
sec_placing_00_reading_0_sp1_1,0,<,sec_placing_00_reading_0_sp1_1
sec_placing_00_reading_0_sp1_1,=,<,fst_placing_00_reading_0_sp1
sec_placing_00_reading_0_sp1_1,1,sec_placing_00_reading_1_sp1
sec_placing_00_reading_1_sp1,=,<,sec_placing_00_reading_1_sp1_1
sec_placing_00_reading_1_sp1_1,1,<,sec_placing_00_reading_1_sp1_1
sec_placing_00_reading_1_sp1_1,0,<,sec_placing_00_reading_1_sp1_1
sec_placing_00_reading_1_sp1_1,=,<,fst_placing_00_reading_1_sp1
// Обработка первого числа
fst_placing_00_reading_0_sp1,1,<,fst_placing_00_reading_0_sp1
fst_placing_00_reading_0_sp1,0,<,fst_placing_00_reading_0_sp1
fst_placing_00_reading_1_sp1,1,<,fst_placing_00_reading_1_sp1
fst_placing_00_reading_1_sp1,0,<,fst_placing_00_reading_1_sp1
// Ставим 0 в первом числе
fst_placing_00_reading_0_sp1,0,fst_placing_00_reading_0_sp1_1_0
fst_placing_00_reading_0_sp1_1_0,0,<,fst_placing_00_reading_0_sp1_1
fst_placing_00_reading_0_sp1_1_0,1,<,fst_placing_00_reading_0_sp1_1
fst_placing_00_reading_1_sp1,0,fst_placing_00_reading_1_sp1_1_0
fst_placing_00_reading_1_sp1_1_0,0,<,fst_placing_00_reading_1_sp1_1
fst_placing_00_reading_1_sp1_1_0,1,<,fst_placing_00_reading_1_sp1_1
// Считывание цифры из первого числа и вызов постановки ответа
fst_placing_00_reading_0_sp1_1_0,go_to_place00_fst_sp0
fst_placing_00_reading_0_sp1_1_1,go_to_place10_fst_sp0
fst_placing_00_reading_1_sp1_1_0,go_to_place01_fst_sp0
fst_placing_00_reading_1_sp1_1_1,go_to_place11_fst_sp0
// Завершение второго числа
sec_placing_00_reading_**_sp1_0,=,sec_placing_00_reading_*.sp1
sec_placing_00_reading_*.sp1,=,sec_placing_00_reading_*.sp1_1
sec_placing_00_reading_*.sp1_1,1,<,sec_placing_00_reading_*.sp1_1
sec_placing_00_reading_*.sp1_1,0,<,sec_placing_00_reading_*.sp1_1
sec_placing_00_reading_*.sp1_1,=,<,fst_placing_00_reading_*.sp1
fst_placing_00_reading_*.sp1,1,<,fst_placing_00_reading_*.sp1
fst_placing_00_reading_*.sp1,0,<,fst_placing_00_reading_*.sp1
// Ставим 0 в первом числе
fst_placing_00_reading_*.sp1,0,fst_placing_00_reading_*.sp1_1_0
fst_placing_00_reading_*.sp1_1_0,0,<,fst_placing_00_reading_*.sp1_1

```

```

fst_placing_00_reading_*.sp1_1_0,1,<,fst_placing_00_reading_*.sp1_1
// Оба числа одинаковой длины окончились
fst_placing_00_reading_*.sp1_1,=,go_to_end
// Последующее возвращение исходных значений + прочтение новых для 01
start01, <,sec_placing_01_reading_**
sec_placing_01_reading_**,1,<,sec_placing_01_reading_**
sec_placing_01_reading_**,0,<,sec_placing_01_reading_**
// Ставим 1 во втором числе
sec_placing_01_reading_**,1,sec_placing_01_reading_**.sp1
sec_placing_01_reading_**.sp1,0,<,sec_placing_01_reading_**.sp1_0
sec_placing_01_reading_**.sp1,1,<,sec_placing_01_reading_**.sp1_0
// Переход к первому числу
sec_placing_01_reading_**.sp1_0,0,sec_placing_01_reading_*.0.sp1
sec_placing_01_reading_*.0.sp1, <,sec_placing_01_reading_*.0.sp1_1
sec_placing_01_reading_*.0.sp1_1,1,<,sec_placing_01_reading_*.0.sp1_1
sec_placing_01_reading_*.0.sp1_1,0,<,sec_placing_01_reading_*.0.sp1_1
sec_placing_01_reading_*.0.sp1_1, <,fst_placing_01_reading_*.0.sp1
sec_placing_01_reading_**.sp1_0,1,sec_placing_01_reading_*.1.sp1
sec_placing_01_reading_*.1.sp1, <,sec_placing_01_reading_*.1.sp1_1
sec_placing_01_reading_*.1.sp1_1,1,<,sec_placing_01_reading_*.1.sp1_1
sec_placing_01_reading_*.1.sp1_1,0,<,sec_placing_01_reading_*.1.sp1_1
sec_placing_01_reading_*.1.sp1_1, <,fst_placing_01_reading_*.1.sp1
// Обработка первого числа
fst_placing_01_reading_*.0.sp1,1,<,fst_placing_01_reading_*.0.sp1
fst_placing_01_reading_*.0.sp1,0,<,fst_placing_01_reading_*.0.sp1
fst_placing_01_reading_*.1.sp1,1,<,fst_placing_01_reading_*.1.sp1
fst_placing_01_reading_*.1.sp1,0,<,fst_placing_01_reading_*.1.sp1
// Ставим 0 в первом числе
fst_placing_01_reading_*.0.sp1,0,fst_placing_01_reading_*.0.sp1_1_0
fst_placing_01_reading_*.0.sp1_1_0,0,<,fst_placing_01_reading_*.0.sp1_1
fst_placing_01_reading_*.0.sp1_1_0,1,<,fst_placing_01_reading_*.0.sp1_1
fst_placing_01_reading_*.1.sp1,0,fst_placing_01_reading_*.1.sp1_1_0
fst_placing_01_reading_*.1.sp1_1_0,0,<,fst_placing_01_reading_*.1.sp1_1
fst_placing_01_reading_*.1.sp1_1_0,1,<,fst_placing_01_reading_*.1.sp1_1
// Считывание цифры из первого числа и вызов постановки ответа
fst_placing_01_reading_*.0.sp1_1_0,go_to_place00_fst_sp0
fst_placing_01_reading_*.0.sp1_1_1,go_to_place10_fst_sp0
fst_placing_01_reading_*.1.sp1_1_0,go_to_place01_fst_sp0
fst_placing_01_reading_*.1.sp1_1_1,go_to_place11_fst_sp0
// Завершение второго числа
sec_placing_01_reading_**.sp1_0,=,sec_placing_01_reading_*.sp1
sec_placing_01_reading_*.sp1,=,sec_placing_01_reading_*.sp1_1
sec_placing_01_reading_*.sp1_1,1,<,sec_placing_01_reading_*.sp1_1
sec_placing_01_reading_*.sp1_1,0,<,sec_placing_01_reading_*.sp1_1
sec_placing_01_reading_*.sp1_1, <,fst_placing_01_reading_*.sp1
fst_placing_01_reading_*.sp1,1,<,fst_placing_01_reading_*.sp1
fst_placing_01_reading_*.sp1,0,<,fst_placing_01_reading_*.sp1
// Ставим 0 в первом числе
fst_placing_01_reading_*.sp1,0,fst_placing_01_reading_*.sp1_1_0
fst_placing_01_reading_*.sp1_1_0,0,<,fst_placing_01_reading_*.sp1_1
fst_placing_01_reading_*.sp1_1_0,1,<,fst_placing_01_reading_*.sp1_1
// Оба числа одинаковой длины окончились
fst_placing_01_reading_*.sp1_1,=,go_to_end
// Последующее возвращение исходных значений + прочтение новых для 10
start10, <,sec_placing_10_reading_**
sec_placing_10_reading_**,1,<,sec_placing_10_reading_**
sec_placing_10_reading_**,0,<,sec_placing_10_reading_**
// Ставим 0 во втором числе
sec_placing_10_reading_**,0,sec_placing_10_reading_**.sp1
sec_placing_10_reading_**.sp1,0,<,sec_placing_10_reading_**.sp1_0
sec_placing_10_reading_**.sp1,1,<,sec_placing_10_reading_**.sp1_0
// Переход к первому числу
sec_placing_10_reading_**.sp1_0,0,sec_placing_10_reading_*.0.sp1
sec_placing_10_reading_*.0.sp1, <,sec_placing_10_reading_*.0.sp1_1
sec_placing_10_reading_*.0.sp1_1,1,<,sec_placing_10_reading_*.0.sp1_1
sec_placing_10_reading_*.0.sp1_1,0,<,sec_placing_10_reading_*.0.sp1_1
sec_placing_10_reading_*.0.sp1_1, <,fst_placing_10_reading_*.0.sp1
sec_placing_10_reading_**.sp1_0,1,sec_placing_10_reading_*.1.sp1
sec_placing_10_reading_*.1.sp1, <,sec_placing_10_reading_*.1.sp1_1

```

```

sec_placing_10_reading_*1_sp1_1,1,<,sec_placing_10_reading_*1_sp1_1
sec_placing_10_reading_*1_sp1_1,0,<,sec_placing_10_reading_*1_sp1_1
sec_placing_10_reading_*1_sp1_1,<,fst_placing_10_reading_*1_sp1
// Обработка первого числа
fst_placing_10_reading_*0_sp1,1,<,fst_placing_10_reading_*0_sp1
fst_placing_10_reading_*0_sp1,0,<,fst_placing_10_reading_*0_sp1
fst_placing_10_reading_*1_sp1,1,<,fst_placing_10_reading_*1_sp1
fst_placing_10_reading_*1_sp1,0,<,fst_placing_10_reading_*1_sp1
// Ставим 1 в первом числе
fst_placing_10_reading_*0_sp1,1,fst_placing_10_reading_*0_sp1_1_0
fst_placing_10_reading_*0_sp1_1_0,0,<,fst_placing_10_reading_*0_sp1_1
fst_placing_10_reading_*0_sp1_1_0,1,<,fst_placing_10_reading_*0_sp1_1
fst_placing_10_reading_*1_sp1,1,fst_placing_10_reading_*1_sp1_1_0
fst_placing_10_reading_*1_sp1_1_0,0,<,fst_placing_10_reading_*1_sp1_1
fst_placing_10_reading_*1_sp1_1_0,1,<,fst_placing_10_reading_*1_sp1_1
// Считывание цифры из первого числа и вызов постановки ответа
fst_placing_10_reading_*0_sp1_1_0,go_to_place00_fst_sp0
fst_placing_10_reading_*0_sp1_1,1,go_to_place10_fst_sp0
fst_placing_10_reading_*1_sp1_1_0,go_to_place01_fst_sp0
fst_placing_10_reading_*1_sp1_1,1,go_to_place11_fst_sp0
// Завершение второго числа
sec_placing_10_reading_**_sp1_0,=,sec_placing_10_reading_*.sp1
sec_placing_10_reading_*.sp1,=,sec_placing_10_reading_*.sp1_1
sec_placing_10_reading_*.sp1_1,1,<,sec_placing_10_reading_*.sp1_1
sec_placing_10_reading_*.sp1_1,0,<,sec_placing_10_reading_*.sp1_1
sec_placing_10_reading_*.sp1_1,<,fst_placing_10_reading_*.sp1
fst_placing_10_reading_*.sp1,1,<,fst_placing_10_reading_*.sp1
fst_placing_10_reading_*.sp1,0,<,fst_placing_10_reading_*.sp1
// Ставим 1 в первом числе
fst_placing_10_reading_*.sp1,1,fst_placing_10_reading_*.sp1_1_0
fst_placing_10_reading_*.sp1_1_0,0,<,fst_placing_10_reading_*.sp1_1
fst_placing_10_reading_*.sp1_1_0,1,<,fst_placing_10_reading_*.sp1_1
// Оба числа одинаковой длины окончились
fst_placing_10_reading_*.sp1_1,=,go_to_end
// Последующее возвращение исходных значений + прочтение новых для 11
start11,<,sec_placing_11_reading_**
sec_placing_11_reading_**,1,<,sec_placing_11_reading_**
sec_placing_11_reading_**,0,<,sec_placing_11_reading_**
// Ставим 1 во втором числе
sec_placing_11_reading_**,1,sec_placing_11_reading_**_sp1
sec_placing_11_reading_**_sp1,0,<,sec_placing_11_reading_**_sp1_0
sec_placing_11_reading_**_sp1,1,<,sec_placing_11_reading_**_sp1_0
// Переход к первому числу
sec_placing_11_reading_**_sp1_0,0,sec_placing_11_reading_*0_sp1
sec_placing_11_reading_*0_sp1,<,sec_placing_11_reading_*0_sp1_1
sec_placing_11_reading_*0_sp1_1,1,<,sec_placing_11_reading_*0_sp1_1
sec_placing_11_reading_*0_sp1_1,0,<,sec_placing_11_reading_*0_sp1_1
sec_placing_11_reading_*0_sp1_1,<,fst_placing_11_reading_*0_sp1
sec_placing_11_reading_**_sp1_0,1,sec_placing_11_reading_*1_sp1
sec_placing_11_reading_*1_sp1,<,sec_placing_11_reading_*1_sp1_1
sec_placing_11_reading_*1_sp1_1,1,<,sec_placing_11_reading_*1_sp1_1
sec_placing_11_reading_*1_sp1_1,0,<,sec_placing_11_reading_*1_sp1_1
sec_placing_11_reading_*1_sp1_1,<,fst_placing_11_reading_*1_sp1
// Обработка первого числа
fst_placing_11_reading_*0_sp1,1,<,fst_placing_11_reading_*0_sp1
fst_placing_11_reading_*0_sp1,0,<,fst_placing_11_reading_*0_sp1
fst_placing_11_reading_*1_sp1,1,<,fst_placing_11_reading_*1_sp1
fst_placing_11_reading_*1_sp1,0,<,fst_placing_11_reading_*1_sp1
// Ставим 1 в первом числе
fst_placing_11_reading_*0_sp1,1,fst_placing_11_reading_*0_sp1_1_0
fst_placing_11_reading_*0_sp1_1_0,0,<,fst_placing_11_reading_*0_sp1_1
fst_placing_11_reading_*0_sp1_1_0,1,<,fst_placing_11_reading_*0_sp1_1
fst_placing_11_reading_*1_sp1,1,fst_placing_11_reading_*1_sp1_1_0
fst_placing_11_reading_*1_sp1_1_0,0,<,fst_placing_11_reading_*1_sp1_1
fst_placing_11_reading_*1_sp1_1_0,1,<,fst_placing_11_reading_*1_sp1_1
// Считывание цифры из первого числа и вызов постановки ответа
fst_placing_11_reading_*0_sp1_1_0,go_to_place00_fst_sp0
fst_placing_11_reading_*0_sp1_1,1,go_to_place10_fst_sp0
fst_placing_11_reading_*1_sp1_1_0,go_to_place01_fst_sp0

```

```

fst_placing_11_reading*_1_sp1_1,1,go_to_place11_fst_sp0
// Завершение второго числа
sec_placing_11_reading**_sp1_0,=,sec_placing_11_reading*._sp1
sec_placing_11_reading*._sp1,=,sec_placing_11_reading*._sp1_1
sec_placing_11_reading*._sp1_1,<,sec_placing_11_reading*._sp1_1
sec_placing_11_reading*._sp1_1,0,<,sec_placing_11_reading*._sp1_1
sec_placing_11_reading*._sp1_1,<,fst_placing_11_reading*._sp1
fst_placing_11_reading*._sp1,1,<,fst_placing_11_reading*._sp1
fst_placing_11_reading*._sp1,0,<,fst_placing_11_reading*._sp1
// Ставим 1 в первом числе
fst_placing_11_reading*._sp1,1,fst_placing_11_reading*._sp1_1_0
fst_placing_11_reading*._sp1_1_0,0,<,fst_placing_11_reading*._sp1_1
fst_placing_11_reading*._sp1_1_0,1,<,fst_placing_11_reading*._sp1_1
// Оба числа одинаковой длины окончились
fst_placing_11_reading*._sp1_1,=,go_to_end
// Первое число закончилось, а второе продолжается
fst_placing_00_reading*_0_sp1_1,>,sec_0
fst_placing_01_reading*_0_sp1_1,>,sec_0
fst_placing_10_reading*_0_sp1_1,>,sec_0
fst_placing_11_reading*_0_sp1_1,>,sec_0
fst_placing_00_reading*_1_sp1_1,>,sec_1
fst_placing_01_reading*_1_sp1_1,>,sec_1
fst_placing_10_reading*_1_sp1_1,>,sec_1
fst_placing_11_reading*_1_sp1_1,>,sec_1
// Ставим результат
sec_0,1,>,sec_0
sec_0,0,>,sec_0
sec_0,>,sec_0_sp1
sec_0_sp1,1,>,sec_0_sp1
sec_0_sp1,0,>,sec_0_sp1
sec_0_sp1,>,sec_0_sp2
sec_0_sp2,1,>,sec_0_sp2
sec_0_sp2,0,>,sec_0_sp2
sec_0_sp2,>,sec_0_sp3
sec_0_sp3,1,>,sec_0_sp3
sec_0_sp3,0,>,sec_0_sp3
sec_0_sp3,0,go_back_sec_0
sec_1,1,>,sec_1
sec_1,0,>,sec_1
sec_1,>,sec_1_sp1
sec_1_sp1,1,>,sec_1_sp1
sec_1_sp1,0,>,sec_1_sp1
sec_1_sp1,>,sec_1_sp2
sec_1_sp2,1,>,sec_1_sp2
sec_1_sp2,0,>,sec_1_sp2
sec_1_sp2,>,sec_1_sp3
sec_1_sp3,1,>,sec_1_sp3
sec_1_sp3,0,>,sec_1_sp3
sec_1_sp3,1,go_back_sec_1
// Постановка и прочтение цифры из второго числа
// Возвращение старого числа
go_back_sec_0,1,<,go_back_sec_0
go_back_sec_0,0,<,go_back_sec_0
go_back_sec_0,<,go_back_sec_0_sp1
go_back_sec_0_sp1,1,<,go_back_sec_0_sp1
go_back_sec_0_sp1,0,<,go_back_sec_0_sp1
go_back_sec_0_sp1,0,go_back_sec_placed_0
go_back_sec_placed_0,1,<,go_back_sec_placed_0_read?
go_back_sec_placed_0,0,<,go_back_sec_placed_0_read?
go_back_sec_1,1,<,go_back_sec_1
go_back_sec_1,0,<,go_back_sec_1
go_back_sec_1,<,go_back_sec_1_sp1
go_back_sec_1_sp1,1,<,go_back_sec_1_sp1
go_back_sec_1_sp1,0,<,go_back_sec_1_sp1
go_back_sec_1_sp1,1,go_back_sec_placed_1
go_back_sec_placed_1,1,<,go_back_sec_placed_1_read?
go_back_sec_placed_1,0,<,go_back_sec_placed_1_read?
//Чтение нового числа
go_back_sec_placed_0_read?,0,place_sec_0

```



```

go_back_sec_placed_0_read?,1, ,place_sec_1
go_back_sec_placed_0_read?, ,=,go_to_end_1
go_back_sec_placed_1_read?,0, ,place_sec_0
go_back_sec_placed_1_read?,1, ,place_sec_1
go_back_sec_placed_1_read?, ,=,go_to_end_1
// Чтение новой цифры и ее запись (сразу начиная со второго числа)
place_sec_0, ,>,place_sec_0_sp1
place_sec_0_sp1,1,>,place_sec_0_sp1
place_sec_0_sp1,0,>,place_sec_0_sp1
place_sec_0_sp1, ,>,place_sec_0_sp2
place_sec_0_sp2,1,>,place_sec_0_sp2
place_sec_0_sp2,0,>,place_sec_0_sp2
place_sec_0_sp2, ,0,go_back_sec_0
place_sec_1, ,>,place_sec_1_sp1
place_sec_1_sp1,1,>,place_sec_1_sp1
place_sec_1_sp1,0,>,place_sec_1_sp1
place_sec_1_sp1, ,>,place_sec_1_sp2
place_sec_1_sp2,1,>,place_sec_1_sp2
place_sec_1_sp2,0,>,place_sec_1_sp2
place_sec_1_sp2, ,1,go_back_sec_1
// Второе число закончилось, а первое продолжается
fst_placing_00_reading*._sp1_1,0, ,sec_ended_fst_0
fst_placing_01_reading*._sp1_1,0, ,sec_ended_fst_0
fst_placing_10_reading*._sp1_1,0, ,sec_ended_fst_0
fst_placing_11_reading*._sp1_1,0, ,sec_ended_fst_0
sec_ended_fst_0, ,>,getres_fst0
fst_placing_00_reading*._sp1_1,1, ,sec_ended_fst_1
fst_placing_01_reading*._sp1_1,1, ,sec_ended_fst_1
fst_placing_10_reading*._sp1_1,1, ,sec_ended_fst_1
fst_placing_11_reading*._sp1_1,1, ,sec_ended_fst_1
sec_ended_fst_1, ,>,getres_fst1
// Получение результата только для 1 числа
getres_fst0, ,>,getres_fst0_sp1 // В этом блоке ставим 0
getres_fst0,1,>,getres_fst0
getres_fst0,0,>,getres_fst0
getres_fst0_sp1, ,>,getres_fst0_place_0
getres_fst0_sp1,1,>,getres_fst0_sp1
getres_fst0_sp1,0,>,getres_fst0_sp1
getres_fst0_place_0, ,0,go_back_fst0
getres_fst0_place_0,1,>,getres_fst0_place_0
getres_fst0_place_0,0,>,getres_fst0_place_0
getres_fst1, ,>,getres_fst1_sp1 // В этом блоке ставим 1
getres_fst1,1,>,getres_fst1
getres_fst1,0,>,getres_fst1
getres_fst1_sp1, ,>,getres_fst1_place_0
getres_fst1_sp1,1,>,getres_fst1_sp1
getres_fst1_sp1,0,>,getres_fst1_sp1
getres_fst1_place_0, ,1,go_back_fst1
getres_fst1_place_0,1,>,getres_fst1_place_0
getres_fst1_place_0,0,>,getres_fst1_place_0
// Возвращаемся, ставим нужное цифру и берем следующую
go_back_fst1,1,<,go_back_fst1
go_back_fst1,0,<,go_back_fst1
go_back_fst1, ,<,go_back_fst1_sp1
go_back_fst1_sp1,1,<,go_back_fst1_sp1
go_back_fst1_sp1,0,<,go_back_fst1_sp1
go_back_fst1_sp1, ,<,go_back_fst1_sp2
go_back_fst1_sp2,1,<,go_back_fst1_sp2
go_back_fst1_sp2,0,<,go_back_fst1_sp2
go_back_fst1_sp2, ,1,go_back_fst?
go_back_fst0,1,<,go_back_fst0
go_back_fst0,0,<,go_back_fst0
go_back_fst0, ,<,go_back_fst0_sp1
go_back_fst0_sp1,1,<,go_back_fst0_sp1
go_back_fst0_sp1,0,<,go_back_fst0_sp1
go_back_fst0_sp1, ,<,go_back_fst0_sp2
go_back_fst0_sp2,1,<,go_back_fst0_sp2
go_back_fst0_sp2,0,<,go_back_fst0_sp2
go_back_fst0_sp2, ,0,go_back_fst?

```

```

// Смотрим, какое число после только что поставленного
// CONST
    go_back_fst?,1,<,go_back_fst?_1
    go_back_fst?,0,<,go_back_fst?_1
    go_back_fst?_1,1, ,getres_fst1_sp0
    getres_fst1_sp0, ,>,getres_fst1
    go_back_fst?_1,0, ,getres_fst0_sp0
    getres_fst0_sp0, ,>,getres_fst0
    go_back_fst?_1, ,=,go_to_end

// Конец
go_to_end, ,>,go_to_end_1
go_to_end,0,>,go_to_end
go_to_end,1,>,go_to_end
go_to_end_1,0,>,go_to_end_1
go_to_end_1,1,>,go_to_end_1
go_to_end_1, ,>,go_to_end_2
go_to_end_2,1,>,go_to_end_2
go_to_end_2,0,>,go_to_end_2
go_to_end_2, ,>,go_to_end_3
go_to_end_3,1,>,go_to_end_3
go_to_end_3,0,>,go_to_end_3
go_to_end_3, ,=,start_reverse_answer
// Проверка, является ли это 1 цифрой
start_reverse_answer, ,<,check_ln1
check_ln1,0,<,check_ln1_sp0
check_ln1,1,<,check_ln1_sp0
check_ln1_sp0, ,>,is_1
is_1,0,>,is_1
is_1,1,>,is_1
check_ln1_sp0,0,>,not_1
check_ln1_sp0,1,>,not_1
not_1,0,>,not_1
not_1,1,>,not_1
is_1, ,#,is_1
not_1, ,<,ln* _rn*
// Условие проверки на предпоследний ход
ln_0_rn_0, ,<,ln_0_rn_0_sp0
ln_0_rn_0_sp0,0,<,ln_0_rn_0_sp0
ln_0_rn_0_sp0,1,<,ln_0_rn_0_sp0
// Два пробела рядом
ln_0_rn_0_sp0, ,<,ln_0_rn_0_sp1
ln_0_rn_0_sp1, ,0,ln_0_rn_0_sp1_placed0*
ln_0_rn_0_sp1_placed0*,0,>,ln_0_rn_0_sp1_placed0*
ln_0_rn_0_sp1_placed0*,1,>,ln_0_rn_0_sp1_placed0*
ln_0_rn_0_sp1_placed0*, ,0,end_0
// Пробел стоит через 1
ln_0_rn_0_sp1,0,<,ln_0_rn_0_sp1_1
ln_0_rn_0_sp1,1,<,ln_0_rn_0_sp1_1
ln_0_rn_0_sp1_1, ,0,ln_0_rn_0_sp1_1_placed0*
ln_0_rn_0_sp1_1_placed0*,0,>,ln_0_rn_0_sp1_1_placed0*
ln_0_rn_0_sp1_1_placed0*,1,>,ln_0_rn_0_sp1_1_placed0*
ln_0_rn_0_sp1_1_placed0*, ,0,end_0
// Возвращение к крайней правой
ln_0_rn_0_sp1_1,0,>,ln_0_rn_0_sp1_0
ln_0_rn_0_sp1_1,1,>,ln_0_rn_0_sp1_0
ln_0_rn_0_sp1_0,0,>,ln_0_rn_0_sp1_0
ln_0_rn_0_sp1_0,1,>,ln_0_rn_0_sp1_0
ln_0_rn_0_sp1_0, ,>,ln_0_rn_0_goright
ln_0_rn_0_goright,0,>,ln_0_rn_0_goright
ln_0_rn_0_goright,1,>,ln_0_rn_0_goright
// Условие проверки на предпоследний ход
ln_0_rn_1, ,<,ln_0_rn_1_sp0
ln_0_rn_1_sp0,0,<,ln_0_rn_1_sp0
ln_0_rn_1_sp0,1,<,ln_0_rn_1_sp0
// Два пробела рядом
ln_0_rn_1_sp0, ,<,ln_0_rn_1_sp1
ln_0_rn_1_sp1, ,1,ln_0_rn_1_sp1_placed0*
ln_0_rn_1_sp1_placed0*,0,>,ln_0_rn_1_sp1_placed0*
ln_0_rn_1_sp1_placed0*,1,>,ln_0_rn_1_sp1_placed0*

```

```

ln_0_rn_1_sp1_placed0*, ,0,end_0
// Пробел стоит через 1
ln_0_rn_1_sp1_0,<,ln_0_rn_1_sp1_1
ln_0_rn_1_sp1_1,<,ln_0_rn_1_sp1_1
ln_0_rn_1_sp1_1, ,1,ln_0_rn_1_sp1_1_placed0*
ln_0_rn_1_sp1_1_placed0*,0,>,ln_0_rn_1_sp1_1_placed0*
ln_0_rn_1_sp1_1_placed0*,1,>,ln_0_rn_1_sp1_1_placed0*
ln_0_rn_1_sp1_1_placed0*, ,0,end_0
// Возвращение к крайней правой
ln_0_rn_1_sp1_1_0,>,ln_0_rn_1_sp1_0
ln_0_rn_1_sp1_1_1,>,ln_0_rn_1_sp1_0
ln_0_rn_1_sp1_0_0,>,ln_0_rn_1_sp1_0
ln_0_rn_1_sp1_0_1,>,ln_0_rn_1_sp1_0
ln_0_rn_1_sp1_0, ,>,ln_0_rn_1_goright
ln_0_rn_1_goright_0,>,ln_0_rn_1_goright
ln_0_rn_1_goright_1,>,ln_0_rn_1_goright
// Условие проверки на предпоследний ход
ln_1_rn_0, ,<,ln_1_rn_0_sp0
ln_1_rn_0_sp0_0,<,ln_1_rn_0_sp0
ln_1_rn_0_sp0_1,<,ln_1_rn_0_sp0
// Два пробела рядом
ln_1_rn_0_sp0, ,<,ln_1_rn_0_sp1
ln_1_rn_0_sp1, ,0,ln_1_rn_0_sp1_placed0*
ln_1_rn_0_sp1_placed0*,0,>,ln_1_rn_0_sp1_placed0*
ln_1_rn_0_sp1_placed0*,1,>,ln_1_rn_0_sp1_placed0*
ln_1_rn_0_sp1_placed0*, ,1,end_0
// Пробел стоит через 1
ln_1_rn_0_sp1_0,<,ln_1_rn_0_sp1_1
ln_1_rn_0_sp1_1,<,ln_1_rn_0_sp1_1
ln_1_rn_0_sp1_1, ,0,ln_1_rn_0_sp1_1_placed0*
ln_1_rn_0_sp1_1_placed0*,0,>,ln_1_rn_0_sp1_1_placed0*
ln_1_rn_0_sp1_1_placed0*,1,>,ln_1_rn_0_sp1_1_placed0*
ln_1_rn_0_sp1_1_placed0*, ,1,end_0
// Возвращение к крайней правой
ln_1_rn_0_sp1_1_0,>,ln_1_rn_0_sp1_0
ln_1_rn_0_sp1_1_1,>,ln_1_rn_0_sp1_0
ln_1_rn_0_sp1_0_0,>,ln_1_rn_0_sp1_0
ln_1_rn_0_sp1_0_1,>,ln_1_rn_0_sp1_0
ln_1_rn_0_sp1_0, ,>,ln_1_rn_0_goright
ln_1_rn_0_goright_0,>,ln_1_rn_0_goright
ln_1_rn_0_goright_1,>,ln_1_rn_0_goright
// Условие проверки на предпоследний ход
ln_1_rn_1, ,<,ln_1_rn_1_sp0
ln_1_rn_1_sp0_0,<,ln_1_rn_1_sp0
ln_1_rn_1_sp0_1,<,ln_1_rn_1_sp0
// Два пробела рядом
ln_1_rn_1_sp0, ,<,ln_1_rn_1_sp1
ln_1_rn_1_sp1, ,1,ln_1_rn_1_sp1_placed0*
ln_1_rn_1_sp1_placed0*,0,>,ln_1_rn_1_sp1_placed0*
ln_1_rn_1_sp1_placed0*,1,>,ln_1_rn_1_sp1_placed0*
ln_1_rn_1_sp1_placed0*, ,1,end_0
// Пробел стоит через 1
ln_1_rn_1_sp1_0,<,ln_1_rn_1_sp1_1
ln_1_rn_1_sp1_1,<,ln_1_rn_1_sp1_1
ln_1_rn_1_sp1_1, ,1,ln_1_rn_1_sp1_1_placed0*
ln_1_rn_1_sp1_1_placed0*,0,>,ln_1_rn_1_sp1_1_placed0*
ln_1_rn_1_sp1_1_placed0*,1,>,ln_1_rn_1_sp1_1_placed0*
ln_1_rn_1_sp1_1_placed0*, ,1,end_0
// Возвращение к крайней правой
ln_1_rn_1_sp1_1_0,>,ln_1_rn_1_sp1_0
ln_1_rn_1_sp1_1_1,>,ln_1_rn_1_sp1_0
ln_1_rn_1_sp1_0_0,>,ln_1_rn_1_sp1_0
ln_1_rn_1_sp1_0_1,>,ln_1_rn_1_sp1_0
ln_1_rn_1_sp1_0, ,>,ln_1_rn_1_goright
ln_1_rn_1_goright_0,>,ln_1_rn_1_goright
ln_1_rn_1_goright_1,>,ln_1_rn_1_goright
// Первое прочтение
ln*_rn*,0, ,ln*_rn0
ln*_rn0, ,<,ln*_rn0_sp1

```

```

ln*_rn0_sp1,1,<,ln*_rn0_sp1
ln*_rn0_sp1,0,<,ln*_rn0_sp1
ln*_rn0_sp1, >,ln*_rn0_sp1_1
ln*_rn0_sp1_1,0, ,ln0_rn0_sp1
ln*_rn0_sp1_1,1, ,ln1_rn0_sp1
ln0_rn0_sp1, >,ln0_rn0_sp2
ln1_rn0_sp1, >,ln1_rn0_sp2
// Возвращение к началу
ln0_rn0_sp2,0,>,ln0_rn0_sp2
ln0_rn0_sp2,1,>,ln0_rn0_sp2
ln0_rn0_sp2, >,ln0_rn0_sp3
ln0_rn0_sp3,0,>,ln0_rn0_sp3
ln0_rn0_sp3,1,>,ln0_rn0_sp3
ln0_rn0_sp3, =,ln_0_rn_0
ln1_rn0_sp2,0,>,ln1_rn0_sp2
ln1_rn0_sp2,1,>,ln1_rn0_sp2
ln1_rn0_sp2, >,ln1_rn0_sp3
ln1_rn0_sp3,0,>,ln1_rn0_sp3
ln1_rn0_sp3,1,>,ln1_rn0_sp3
ln1_rn0_sp3, =,ln_1_rn_0
ln0_rn1_sp2,0,>,ln0_rn1_sp2
ln0_rn1_sp2,1,>,ln0_rn1_sp2
ln0_rn1_sp2, >,ln0_rn1_sp3
ln0_rn1_sp3,0,>,ln0_rn1_sp3
ln0_rn1_sp3,1,>,ln0_rn1_sp3
ln0_rn1_sp3, =,ln_0_rn_1
ln1_rn1_sp2,0,>,ln1_rn1_sp2
ln1_rn1_sp2,1,>,ln1_rn1_sp2
ln1_rn1_sp2, >,ln1_rn1_sp3
ln1_rn1_sp3,0,>,ln1_rn1_sp3
ln1_rn1_sp3,1,>,ln1_rn1_sp3
ln1_rn1_sp3, =,ln_1_rn_1
ln*_rn*,1, ,ln*_rn1
ln*_rn1, <,ln*_rn1_sp1
ln*_rn1_sp1,1,<,ln*_rn1_sp1
ln*_rn1_sp1,0,<,ln*_rn1_sp1
ln*_rn1_sp1, >,ln*_rn1_sp1_1
ln*_rn1_sp1_1,0, ,ln0_rn1_sp1
ln*_rn1_sp1_1,1, ,ln1_rn1_sp1
ln0_rn1_sp1, >,ln0_rn1_sp2
ln1_rn1_sp1, >,ln1_rn1_sp2
// После первого прочтения число нельзя перевернуть:
// Для 00
ln_0_rn_0_goright, <,readed_00_reading_**
readed_00_reading_**,1,<,readed_00_reading_**
readed_00_reading_**,0,<,readed_00_reading_**
readed_00_reading_**, ,0,readed_+0_reading_**
readed_+0_reading_**,0,<,readed_+0_reading_**_1
readed_+0_reading_**,1,<,readed_+0_reading_**_1
// Справа 0
readed_+0_reading_**_1,0, ,readed_+0_reading_**0
readed_+0_reading_**0, <,readed_+0_reading_**0_sp1
readed_+0_reading_**0_sp1,1,<,readed_+0_reading_**0_sp1
readed_+0_reading_**0_sp1,0,<,readed_+0_reading_**0_sp1
readed_+0_reading_**0_sp1, ,0,readed_++_reading_**0_sp1
readed_++_reading_**0_sp1,0,>,readed_++_reading_**0_sp1_1
readed_++_reading_**0_sp1,1,>,readed_++_reading_**0_sp1_1
readed_++_reading_**0_sp1_1,0, ,ln0_rn0_sp1
readed_++_reading_**0_sp1_1,1, ,ln1_rn0_sp1
// Справа 1
readed_+0_reading_**_1,1, ,readed_+0_reading_**1
readed_+0_reading_**1, <,readed_+0_reading_**1_sp1
readed_+0_reading_**1_sp1,1,<,readed_+0_reading_**1_sp1
readed_+0_reading_**1_sp1,0,<,readed_+0_reading_**1_sp1
readed_+0_reading_**1_sp1, ,0,readed_++_reading_**1_sp1
readed_++_reading_**1_sp1,0,>,readed_++_reading_**1_sp1_1
readed_++_reading_**1_sp1,1,>,readed_++_reading_**1_sp1_1
readed_++_reading_**1_sp1_1,0, ,ln0_rn1_sp1
readed_++_reading_**1_sp1_1,1, ,ln1_rn1_sp1

```

```
// Для 01
ln_0_rn_1_goright, <,readed_01_reading_**
readed_01_reading_**,1,<,readed_01_reading_**
readed_01_reading_**,0,<,readed_01_reading_**
readed_01_reading_**,0,readed_+1_reading_**
readed_+1_reading_**,0,<,readed_+1_reading_**_1
readed_+1_reading_**,1,<,readed_+1_reading_**_1
// Справа 0
readed_+1_reading_**_1,0,readed_+1_reading_**0
readed_+1_reading_**0,<,readed_+1_reading_**0_sp1
readed_+1_reading_**0_sp1,1,<,readed_+1_reading_**0_sp1
readed_+1_reading_**0_sp1,0,<,readed_+1_reading_**0_sp1
readed_+1_reading_**0_sp1,,1,readed_++_reading_**0_sp1
// Справа 1
readed_+1_reading_**_1,1,readed_+1_reading_**1
readed_+1_reading_**1,<,readed_+1_reading_**1_sp1
readed_+1_reading_**1_sp1,1,<,readed_+1_reading_**1_sp1
readed_+1_reading_**1_sp1,0,<,readed_+1_reading_**1_sp1
readed_+1_reading_**1_sp1,,1,readed_++_reading_**1_sp1
// Для 10
ln_1_rn_0_goright, <,readed_10_reading_**
readed_10_reading_**,1,<,readed_10_reading_**
readed_10_reading_**,0,<,readed_10_reading_**
readed_10_reading_**,1,readed_+0_reading_**
// Для 11
ln_1_rn_1_goright, <,readed_11_reading_**
readed_11_reading_**,1,<,readed_11_reading_**
readed_11_reading_**,0,<,readed_11_reading_**
readed_11_reading_**,1,readed_+1_reading_**
// КОНЕЦ
end_0,0,>,end_0
end_0,1,>,end_0
end_0, #,end_0
end, #,end
```

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.

№	Ла б. ил и до м.	Дата	Врем я	Событие	Действие по исправлению	Примечание
1	дом	28.09.22	16:00	Удалил нечаянно программу	Начал писать снова	
2	дом	28.09.22	19:00	Снова нечаянно стер часть программы Нечаянно стер часть кода, которая	Восстановил стертую часть	
3	дом	30.09.22	01:00	переворачивает число	Восстановил стертую часть	

10. Замечания автора по существу работы

11. Выводы

На самом деле работа мне показалась интересной, как и прошлая лабораторная работа с bash, она заставила меня сильно подумать и посидеть над ее решением. Писал я ее больше 3-х дней. Делая эту работы мне было тяжело, однако я справился, и, считаю, что стал немного умнее.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента
