

Московский авиационный институт  
(национально исследовательский университет)

Курсовая работа по курсу:  
«Фундаментальная информатика»

1 семестр

Задание №1

Тема: «Работа с машиной Тьюринга»

Преподаватель: доцент кафедры 806 Никулин С.П.

Студент: Бугренков Владимир Петрович

Группа: М80-111Б-23

г. Москва 2023

## остановка задачи

Вычисление поразрядной конъюнкции двух двоичных чисел (Вариант №1).

## общий метод решения

Программа будет дописывать справа числа, которые будут являться результатом поразрядной конъюнкции двух двоичных чисел, согласно следующему алгоритму: если первая цифра первого числа не равна «0» и первая цифра второго числа не равна «0», то дописываем справа «1», иначе дописываем «0», потом то же самое делает со вторыми, третьими, ... , последними цифрами чисел.

## общие сведения о программе

Программа выполнялась на эмуляторе машины Тьюринга jstu4. Выполнение отчёта происходило при помощи инструмента turun и операционной системы семейства Unix Ubuntu версии 23.04.

## ограничения на объём и величину обрабатываемых данных

Объём и величина обрабатываемых данных не ограничены. Входные данные должны быть записаны в двоичной системе счисления, длина чисел должна совпадать, то есть оба числа должны иметь одинаковое количество разрядов. В случае, если первое число меньше второго, для него нужно слева дописать незначащие нули так, чтобы длина чисел была одинаковой и программа работала корректно. (Например, заданы числа 11111 и 111, для корректной работы программы следует в ленту вписать следующее: « 11111

## ограничения на объём и величину обрабатываемых данных

Сначала головка идет влево до пробела перед первым числом,

После головка идет вправо до первой цифры числа:

*начинается первое ветвление)*

- если первая цифра первого числа будет равна «1», то программа заменит её на «a» и затем головка переместится к первой цифре второго числа: *#(начинается второе ветвление)*

если первая цифра второго числа равна «1», то программа заменит её на «a» и головка переместится к концу второго числа и через пробел вставит «1»

если первая цифра второго числа равна «0», то программа

заменит её на «b» и головка переместиться к концу второго числа и через пробел вставит «0»

- если первая цифра первого числа будет равна «0», то программа заменит её на «b» и затем головка переместиться к первой цифре второго числа: *#(начинается второе ветвление)*

если первая цифра второго числа равна «1», то программа заменит её на «a» и головка переместиться к концу второго числа и через пробел вставит «0»

если первая цифра второго числа равна «0», то программа заменит её на «b» и головка переместиться к концу второго числа и через пробел вставит «0»

*Алгоритм*) После программа вернется к началу первого числа: если символ равен «0» или «1», то головка перемещается вправо, пока не встретит символ «a» или «b». Если символ равен «a», то программа заменяет его на «1», если символ равен «b», то заменит на «0», и после головка перемещается к следующему символу:

*(начинается первое ветвление)*

если следующий символ *#(работаем еще с первым числом)* равен «1», то заменяет её на «a» и головка перемещается к началу второго числа, после головка будет перемещаться по числу вправо, пока не встретит «a» или «b». Если встретит «a», то заменит на «1», если встретит «b», то заменит на «0». После перемещается к следующему символу второго числа: *#(начинается второе ветвление)*

- если символ равен «1», то заменяет на «a» и идёт вправо до конца и дописывает «1»
- если символ равен «0», то заменяет на «b» и идёт вправо до конца и дописывает «0»

если следующий символ *#(работаем еще с первым числом)* равен «0», то заменяет её на «a» и головка перемещается к началу второго числа, после головка будет перемещаться по числу вправо, пока не встретит «a» или «b». Если встретит «a», то заменит на «1», если встретит «b», то заменит на «0». После перемещается к следующему символу второго числа:

*начинается второе ветвление)*

- если символ равен «1», то заменяет на «a» и идёт вправо до конца и дописывает «0»
- если символ равен «0», то заменяет на «b» и идёт вправо до конца и дописывает «0»

если следующий символ *#(работаем еще с первым числом)* равен пробелу « », то головка идет вправо до конца и завершает работу!

- После программа будет возвращаться в начало первого числа и выполнять *алгоритм*, пока не завершит работу

### писание подпрограммы

Подпрограмм нет.

### ходные данные

Два двоичных числа, записанных через пробел, при два числа имеют одинаковое количество разрядов.

### ыходные данные

Два заданных двоичных числа и через пробел число, которое является поразрядной конъюнкции двух заданных двоичных чисел.

### естовые примеры

=> 10 11 10

=> 1111 1001 1001

=> 000111 111011 000011

=> 111 001 001

0

1 => 10 01 00

### невник отладки

root @qwental-VirtualBox:/mnt/c/torun# ./turun mt.tu test.tst

00, <,01	03,a,>,04	06,0,>,06	09,b,<,10	12,b,>,18
01,1,<,01	03, >,33	06,1,>,06	09, <,09	12,1,a,12
01,0,<,01	04,0,>,04	06, >,07	10,0,<,10	12,a,>,13
01, <,02	04,1,>,04	07,0,>,07	10,1,<,10	12, >,16
02,1,<,02	04, >,05	07,1,>,07	10,a,1,11	13,0,>,13
02,0,<,02	05,0,b,05	07, ,1,08	11,1,>,12	13,1,>,13
02, >,03	05,b,>,22	08,1,<,09	10,b,0,11	13,a,1,14
03,0,b,03	05,1,a,05	09,0,<,09	11,0,>,25	14,1,>,15
03,b,>,18	05,a,>,06	09,1,<,09	10, <,10	13,b,0,14
03,1,a,03	05, , ,33	09,a,<,10	12,0,b,12	14,0,>,05

13, >,13	18,1,>,18	22,0,>,22	25, >,31	30,a,>,06
15,0,b,15	18,0,>,18	22,1,>,22	27,0,>,27	31,0,>,31
15,b,>,22	18,a,1,19	22, >,23	27,1,>,27	31,1,>,31
15,1,a,15	19,1,>,20	23,0,>,23	27,a,1,28	31,a,1,32
15,a,>,06	18,b,0,19	23,1,>,23	28,1,>,29	32,1,>,33
16,0,>,16	19,0,>,20	23, ,0,24	27,b,0,28	31,b,0,32
16,1,>,16	18, >,18	24,0,>,09	28,0,>,29	32,0,>,33
16,a,1,17	20,0,b,21	25,0,b,26	27, >,27	33, >,34
17,1,>,33	21,b,>,22	26,b,>,18	29,0,b,30	34,1,>,34
16,b,0,17	20,1,a,21	25,1,a,26	30,b,>,22	34,0,>,34
17,0,>,33	21,a,>,22	26,a,>,27	29,1,a,30	34, , ,34

10 11 ==>

a0 11 ==>

a0 a1 ==>

a0 a1 1 ==>

a0 a1 1 ==>

10 a1 1 ==>

1b a1 1 ==>

1b 11 1 ==>

1b 1a 1 ==>

1b 1a 10 ==>

10 1a 10 ==>

10 11 10 ==>

10 11 10 ==>

10 11 10

Machine stopped successfully

1111 1001 ==>

a111 1001 ==>

a111 a001	=>
a111 a001 1	=>
1111 a001 1	=>
1a11 a001 1	=>
1a11 1001 1	=>
1a11 1b01 1	=>
1a11 1b01 10	=>
1111 1b01 10	=>
11a1 1b01 10	=>
11a1 1001 10	=>
11a1 10b1 10	=>
11a1 10b1 100	=>
1111 10b1 100	=>
111a 10b1 100	=>
111a 1001 100	=>
111a 100a 100	=>
111a 100a 1001	=>
1111 100a 1001	=>
1111 1001 1001	=>
1111 1001 1001	

Machine stopped successfully

000111 111011	=>
b00111 111011	=>
b00111 a11011	=>
b00111 a11011	=>
b00111 a11011 0	=>
b00111 a11011 0	=>
000111 a11011 0	=>
0b0111 a11011 0	=>

0b0111 111011 0	=>
0b0111 1a1011 0	=>
0b0111 1a1011 00	=>
000111 1a1011 00	=>
00b111 1a1011 00	=>
00b111 111011 00	=>
00b111 11a011 00	=>
00b111 11a011 000	=>
000111 11a011 000	=>
000a11 11a011 000	=>
000a11 111011 000	=>
000a11 111b11 000	=>
000a11 111b11 0000	=>
000111 111b11 0000	=>
0001a1 111b11 0000	=>
0001a1 111011 0000	=>
0001a1 1110a1 0000	=>
0001a1 1110a1 00001	=>
000111 1110a1 00001	=>
00011a 1110a1 00001	=>
00011a 111011 00001	=>
00011a 11101a 00001	=>
00011a 11101a 000011	=>
000111 11101a 000011	=>
000111 111011 000011	=>
000111 111011 000011	

Machine stopped successfully

111 001	=>
a11 001	=>

a11 b01 =>

a11 b01 0 =>

111 b01 0 =>

1a1 b01 0 =>

1a1 001 0 =>

1a1 0b1 0 =>

1a1 0b1 00 =>

111 0b1 00 =>

11a 0b1 00 =>

11a 001 00 =>

11a 00a 00 =>

11a 00a 001 =>

111 00a 001 =>

111 001 001 =>

111 001 001

Machine stopped successfully

10 01 =>

a0 01 =>

a0 b1 =>

a0 b1 0 =>

10 b1 0 =>

1b b1 0 =>

1b 01 0 =>

1b 0a 0 =>

1b 0a 00 =>

10 0a 00 =>

10 01 00 =>

10 01 00

Machine stopped successfully



### **ывод по работе**

Поставленная задача была выполнена, в процессе выполнения работы я научился работать с машиной Тьюринга.