Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

ПИиКТ

Лабораторная работа 3

по дисциплине

«Архитектура компьютера»

Выполнили: Студенты группы P33113

Мансуров Б.Б.

Преподаватель: Тропченко А.Ю.

Санкт-Петербург

2020г

# Задание

1. Упорядочить текст лексикографически, в порядке возрастания ASCII- кода

“This programmator”  “aaghimmootTrrs”

1. Вставить пробелы после символа “r”

“This programmator”  ”r”  “This pr ogr ammator”

1. Заменить прописную букву “x” на заглавную в тексте

“This programmator” ”a”  “This progrAmmAtor”

1. Символьное (в ASCII) преобразование двоичного числа в шестнадцатеричное

“01001001110”  “0x24e”

1. Символьное (в ASCII) преобразование шестнадцатеричного числа в двоичное

“01001001110”  “0x24e”

6. Символьное двоичное число преобразовать в символьное десятичное.

“123” 🡨 (“01110101“ )2

7.Символьное (в ASCII) преобразование десятичного числа в шестнадцатеричное

“ 590 ”  “0x24e”

8.Символьное (в ASCII) преобразование шестнадцатеричного числа в десятичное

“590”  “0x24e”

9.Преобразовать число с естественной запятой в полулогарифмическую форму в десятичной системе с учетом знака порядка и знака мантиссы

“-25,023”  “e+2 - 0.25023”

10. Десятичное сложение (вычитание) в неупакованных форматах, положение запятой фиксировано

“256,54” +” 75, 56” =” 332,10”

11. Преобразовать символьный двоичный код в символьный Манчестерский код и восстановить исходный двоичный

“01011010”  00 11 00 11 11 00 11 00 (+)

10 10 10 10 10 10 10 10 синхросигнал

 “10 01 10 01 01 10 01 10 “ Манчестерский код

Восстановление символьного двоичного кода из Манчестерского

“1001100101100110” Манчестерский код

 “ 0 1 0 1 1 0 1 0” двоичный код

12. Сформировать сдачу минимальным количеством монет достоинством **50, 10, 5, 1** копеек и проверить обратным преобразованием

“132”  “2 ,3,0, 2”

13. Шифрование и дешифрование Гронсфельда

таблица символов {a,b,c,d,e,f, …}

нумерация 0 1 2 3 4 5 6

ключ {3,1,2,0,6, …}

Пример

“cadda”  “ cdaad”

14. Преобразование двоичной импульсной последовательности в 3-значный код, перепад 0/1 обозначается 1, 1/0 обозначается 2, отсутствие перепада – 0 и обратно

“0 1 0 0 0 1 0 1 1“  “2 1 0 0 2 1 2 0”

15. Байты данных разбиваются на 2 тетрады, каждая тетрада заменяется HEX-цифрой и преобразуется в ASCII-код, подсчет контрольной суммы байтов по модулю 0x100 в конце строки HEX-кода

A0, B1, 0C, 1D  HEX-код строки “ A 0 B 1 0 C 1 D 8 A”

16.Обратное преобразование HEX-кода в строку байтов данных и проверить контрольную сумму - последний байт в строке

A0, B1, 0C, 1D  “ A 0 B 1 0 C 1 D 8 A” HEX-код строки

17. Регистр граничного сканирования n-контактов в JTAG-интерфейсе имеет длину 3n бит. Выбрать 3-хбитную i-ую ячейку в регистре. Нумерация битов регистра справа налево 3n,…2,1,0

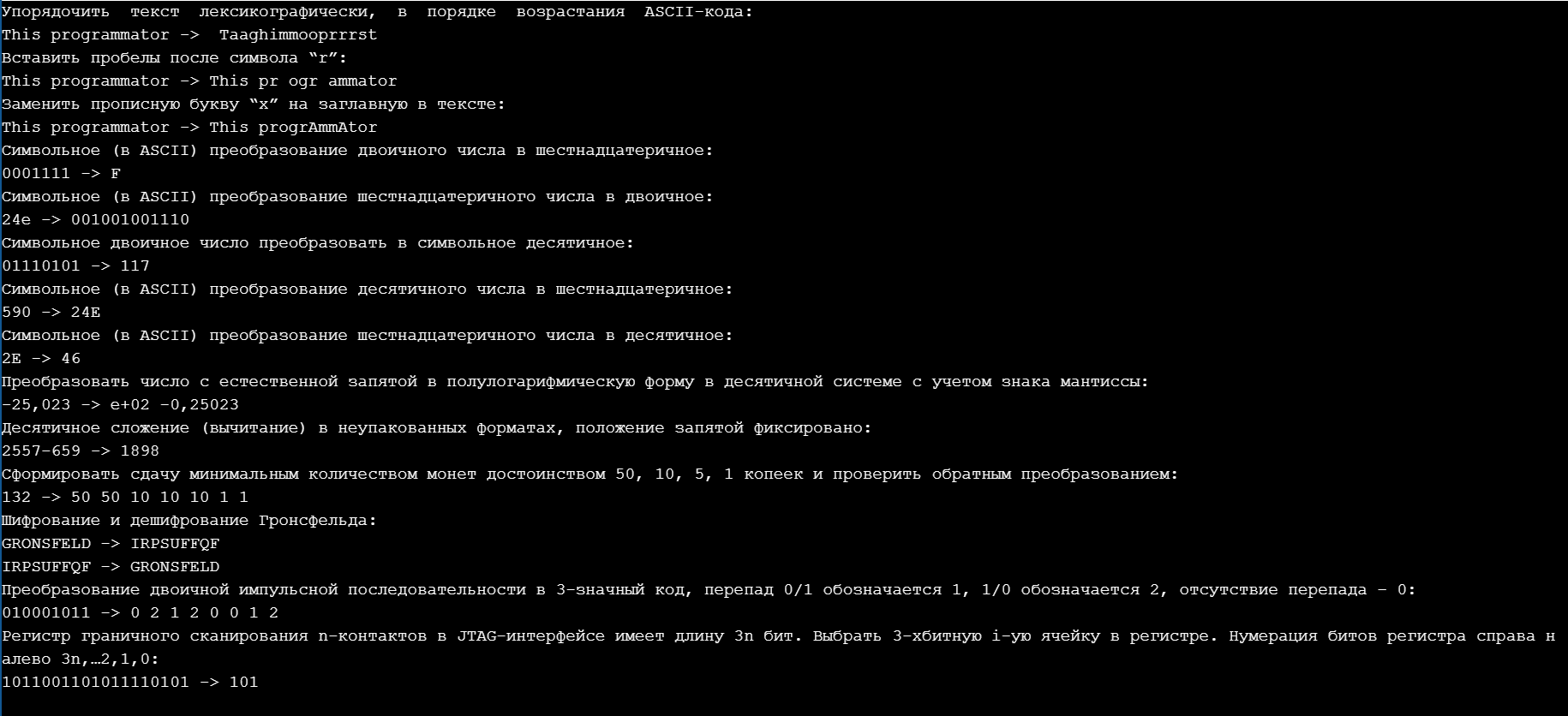
“1 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0 1 0 1” 🡪 “100”

o c i i c o

# Исходный код

<https://github.com/MansurovB-source>

# Вывод программы



# Вывод

Изучил основы языка С и реализовал несколько полезных функций.