

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

Лабораторная работа 3
по дисциплине
«Архитектура компьютера»

Выполнили: Студенты группы Р33113

Мансуров Б.Б.

Преподаватель: Тропченко А.Ю.

Санкт-Петербург

2020г

Задание

1. Упорядочить текст лексикографически, в порядке возрастания ASCII- кода
"This programmer" \rightarrow "aaghimootTrrs"
2. Вставить пробелы после символа "r"
"This programmer" \rightarrow "r" \rightarrow "This pr ogr amimator"
3. Заменить прописную букву "x" на заглавную в тексте
"This programmer" \rightarrow "a" \rightarrow "This progrAmmAtor"
4. Символьное (в ASCII) преобразование двоичного числа в шестнадцатеричное
"01001001110" \rightarrow "0x24e"
5. Символьное (в ASCII) преобразование шестнадцатеричного числа в двоичное
"01001001110" \leftarrow "0x24e"
6. Символьное двоичное число преобразовать в символьное десятичное.
"123" \leftarrow ("01110101")₂
- 7.Символьное (в ASCII) преобразование десятичного числа в шестнадцатеричное
" 590 " \rightarrow "0x24e"
- 8.Символьное (в ASCII) преобразование шестнадцатеричного числа в десятичное
"590" \leftarrow "0x24e"
- 9.Преобразовать число с естественной запятой в полулогарифмическую форму в десятичной системе с учетом знака порядка и знака мантиссы
"-25,023" \rightarrow "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" \rightarrow 00 11 00 11 11 00 11 00 (+)

10 10 10 10 10 10 10 10 синхросигнал

\rightarrow "10 01 10 01 01 10 01 10 " Манчестерский код

Восстановление символьного двоичного кода из Манчестерского

"1001100101100110" Манчестерский код

\rightarrow "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

$$"132" \rightarrow "2,3,0,2"$$

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" \leftrightarrow "cdaad"

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

$$"010001011" \leftrightarrow "21002120"$$

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 \rightarrow HEX-код строки "A0B10C1D8A"

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 0 1 0 1" → "100"

o s i i s o

Исходный код

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

Вывод программы

```
Упорядочить текст лексикографически, в порядке возрастания ASCII-кода:
This programmer -> Taaghimmoorrrst
Вставить пробелы после символа "r":
This programmer -> This pr ogr ammat or
Заменить прописную букву "X" на заглавную в тексте:
This programmer -> This progrAmMAtor
Символьное (в ASCII) преобразование двоичного числа в шестнадцатеричное:
0001111 -> F
Символьное (в ASCII) преобразование шестнадцатеричного числа в двоичное:
24e -> 001001001110
Символьное двоичное число преобразовать в символьное десятичное:
01110101 -> 117
Символьное (в ASCII) преобразование десятичного числа в шестнадцатеричное:
590 -> 24E
Символьное (в ASCII) преобразование шестнадцатеричного числа в десятичное:
2E -> 46
Преобразовать число с естественной запятой в полулогарифмическую форму в десятичной системе с учетом знака мантиссы:
-25,023 -> e+02 -0,25023
Десятичное сложение (вычитание) в неупакованных форматах, положение запятой фиксировано:
2557-659 -> 1898
Сформировать сдачу минимальным количеством монет достоинством 50, 10, 5, 1 копеек и проверить обратным преобразованием:
132 -> 50 50 10 10 10 1 1
Шифрование и дешифрование Гронсфелда:
GRONSFELD -> IRPSUFFQF
IRPSUFFQF -> GRONSFELD
Преобразование двоичной импульсной последовательности в 3-значный код, перепад 0/1 обозначается 1, 1/0 обозначается 2, отсутствие перепада - 0:
010001011 -> 0 2 1 2 0 0 1 2
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:
1011001101011110101 -> 101
```

Вывод

Изучил основы языка C и реализовал несколько полезных функций.