



УНИВЕРСИТЕТ ИТМО

Факультет ПИиКТ

Лабораторная работа №5 по Основам профессиональной  
деятельности

Дополнительное задание

Программа, которая переводит строку в КОИ-8 в строку UTF-8

**Выполнил:**

Давааням Баясгалан

группа Р3111

**Преподаватель:**

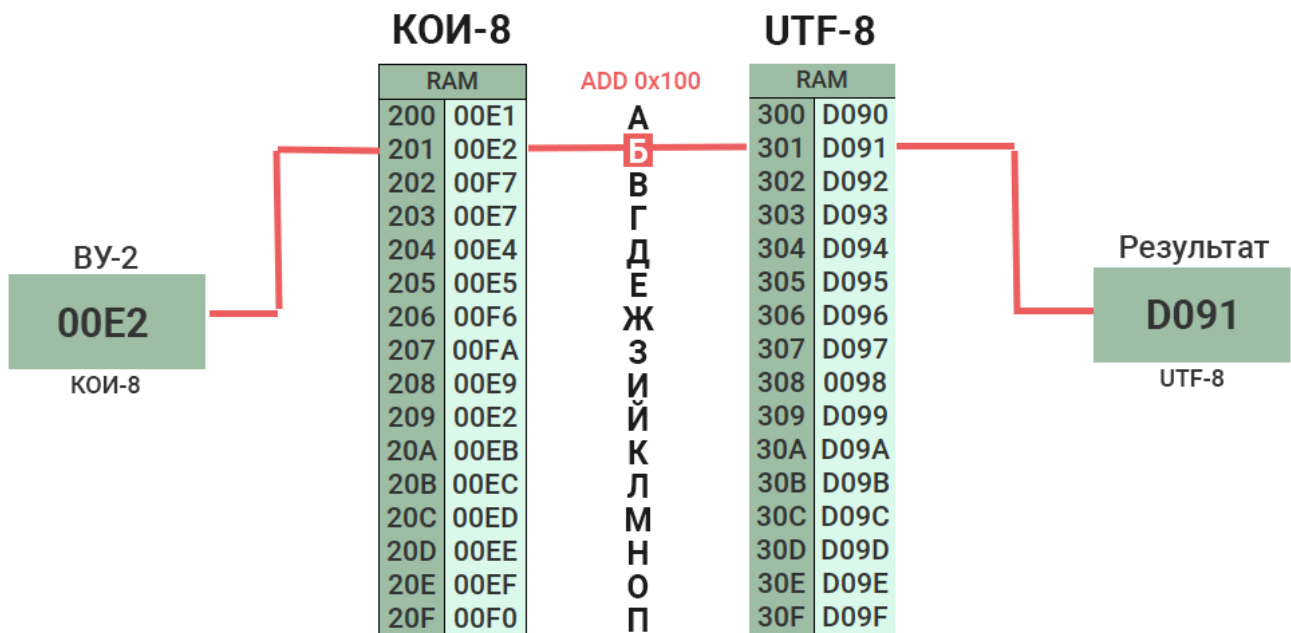
Саржевский Иван Анатольевич

г. Санкт-Петербург

2022 год

В КОИ-8 кириллические буквы стоят беспорядочно относительно UTF-8, поэтому не нашел конкретный алгоритм.

Пришел к выводу, что переводить строку способом DICTIONARY.



Тест

INPUT: Привет! = (F0 D2 C9 D7 C5 D4 21)

Результат в память

RAM	
400	D09F
401	D180
402	D0B8
403	D0B2
404	D0B5
405	D182
406	0021
407	00FF

The screenshot shows a hex editor interface with a 'hexadecimal' tab selected. The input string 'Привет !' is displayed in the 'utf8' tab. The hex editor shows the following data:

Address	Hex
00000000	d09fd180d0b8d0b2d0b5d1820021

The interface includes buttons for 'Import from file', 'Save as...', and 'Copy to clipboard'.

## Код на ассемблере:

```
ORG 0x100
CURRENT: WORD 0x000           // Ячейка, в которой сохраняется текущий элемент
KOI8: WORD 0x200             // Ячейка, в которой сохраняется адрес начала массива элементов KOI-8
UTF8: WORD 0x300             // Ячейка, в которой сохраняется адрес начала массива элементов UTF-8
INDEX_ARRAY: WORD 0x000      // Ячейка, в которой сохраняется проверяющий адрес
INCREASE: WORD 0x0FF         // Ячейка, в которой сохраняется измерение адреса между массивами KOI-8 и UTF-8
RESULT: WORD 0x400           // Ячейка, в которой сохраняется адрес начало результата
MAXIMUM: WORD 0x80           // Ячейка, в которой сохраняется предел между латинскими и кириллическими буквами
NUMBER_LOOP: WORD 0x40       // Количество цикл, который проверяет элементы
STOP: WORD 0xFF              // СТОП символ
INPUT: IN 5                  // Подпрограмма INPUT
    AND #0x40                // Считывать с ВУ-2 и сохранить результат в стек
    BEQ INPUT
    IN 4
    ST (SP+1)
    RET                      // Конец подпрограммы
CHECK_KOI8: LD KOI8          // Подпрограмма CHECK_KOI8
    ST INDEX_ARRAY           // Главное - это подпрограмма найдет значение UTF-8, которое
    LD CURRENT               // подходит значение KOI-8.
COMPARE: CMP (INDEX_ARRAY)+  // 1) Найти значение KOI-8, которое считывал с ВУ-2 с помощью
    BNE ARRAY                // цикла из адреса 0x200 до NUMBER_LOOP
    LD START_MASSIVE         // 2) После того как нашел значение, добавит в текущий адрес INCREASE
    ADD INCREASE              // и сразу переходит в подходящий адрес, в котором сохраняется значение UTF-8.
    ST START_MASSIVE         // 3) Значение UTF-8 сохраняется в CURRENT
    LD (INDEX_ARRAY)         // 4) Если вообще не найдено значение KOI-8, то возвращается значение 0x0000.
    ST CURRENT
    RET
ARRAY: LOOP NUMBER_LOOP
    JUMP COMPARE
    LD 0x0
    ST CURRENT
    RET                      // Конец подпрограммы
START: CLA                   // НАЧАЛО ПРОГРАММЫ
FOR: PUSH
    CALL INPUT                // Считываем СИМВОЛ
    POP
    ST CURRENT
    CMP STOP                  // Проверяет, что он СТОП символ.
    BEQ EXIT
    SUB MAXIMUM               // Проверяет, что он Латинская буква
    BMI EXIT_1
    CALL CHECK_KOI8           // Если он кириллическая буква, то вызывает подпрограмму CHECK_KOI8
EXIT_1: LD CURRENT
    ST (RESULT)+              // Сохраняет полученный результат в соответствующий адрес
    JUMP START
EXIT: ST (RESULT)+
    HLT
ORG 0x200                    // НАЧАЛО МАССИВА ЭЛЕМЕНТОВ KOI-8
Capital_A: WORD 0x00E1
Capital_Be: WORD 0x00E2
Capital_Ve: WORD 0x00F7
Capital_Ghe: WORD 0x00E7
Capital_De: WORD 0x00E4
Capital_Ie: WORD 0x00E5
Capital_Zhe: WORD 0x00F6
Capital_Ze: WORD 0x00FA
Capital_I: WORD 0x00E9
Capital_ShortI: WORD 0x00E2
Capital_Ka: WORD 0x00EB
Capital_El: WORD 0x00EC
Capital_Em: WORD 0x00ED
Capital_En: WORD 0x00EE
Capital_O: WORD 0x00EF
Capital_Pe: WORD 0x00F0
Capital_Er: WORD 0x00F2
Capital_Es: WORD 0x00F3
Capital_Te: WORD 0x00F4
Capital_U: WORD 0x00F5
Capital_Ef: WORD 0x00E6
Capital_Ha: WORD 0x00E8
```

```
Capital_Tse: WORD 0x00E3
Capital_Che: WORD 0x00FE
Capital_Sha: WORD 0x00FB
Capital_Shcha: WORD 0x00FD
Capital_Yeru: WORD 0x00F9
Capital_SoftSign: WORD 0x00F8
Capital_E: WORD 0x00FC
Capital_Yu: WORD 0x00E0
Capital_Ya: WORD 0x00F1
Small_A: WORD 0x00C1
Small_Be: WORD 0x00C2
Small_Ve: WORD 0x00D7
Small_Ghe: WORD 0x00C7
Small_De: WORD 0x00C4
Small_Ie: WORD 0x00C5
Small_Zhe: WORD 0x00D6
Small_Ze: WORD 0x00DA
Small_I: WORD 0x00C9
Small_ShortI: WORD 0x00CA
Small_Ka: WORD 0x00CB
Small_El: WORD 0x00CC
Small_Em: WORD 0x00CD
Small_En: WORD 0x00CE
Small_O: WORD 0x00CF
Small_Pe: WORD 0x00D0
Small_Er: WORD 0x00D2
Small_Es: WORD 0x00D3
Small_Te: WORD 0x00D4
Small_U: WORD 0x00D6
Small_Ef: WORD 0x00C6
Small_Ha: WORD 0x00C8
Small_Tse: WORD 0x00C3
Small_Che: WORD 0x00DE
Small_Sha: WORD 0x00DB
Small_Shcha: WORD 0x00DD
Small_HardSign: WORD 0x00DF
Small_Yeru: WORD 0x00D9
Small_SoftSign: WORD 0x00D8
Small_E: WORD 0x00C5
Small_Yu: WORD 0x00C0
Small_Ya: WORD 0x00D1          // КОНЕЦ МАССИВА ЭЛЕМЕНТОВ КОИ-8
ORG 0x300                     // НАЧАЛО МАССИВА ЭЛЕМЕНТОВ UTF-8
UxCapital_A: WORD 0xD090
UxCapital_Be: WORD 0xD091
UxCapital_Ve: WORD 0xD092
UxCapital_Ghe: WORD 0xD093
UxCapital_De: WORD 0xD094
UxCapital_Ie: WORD 0xD095
UxCapital_Zhe: WORD 0xD096
UxCapital_Ze: WORD 0xD097
UxCapital_I: WORD 0xD098
UxCapital_ShortI: WORD 0xD099
UxCapital_Ka: WORD 0xD09A
UxCapital_El: WORD 0xD09B
UxCapital_Em: WORD 0xD09C
UxCapital_En: WORD 0xD09D
UxCapital_O: WORD 0xD09E
UxCapital_Pe: WORD 0xD09F
UxCapital_Er: WORD 0xD0A0
UxCapital_Es: WORD 0xD0A1
UxCapital_Te: WORD 0xD0A2
UxCapital_U: WORD 0xD0A3
UxCapital_Ef: WORD 0xD0A4
UxCapital_Ha: WORD 0xD0A5
UxCapital_Tse: WORD 0xD0A6
UxCapital_Che: WORD 0xD0A7
UxCapital_Sha: WORD 0xD0A8
UxCapital_Shcha: WORD 0xD0A9
UxCapital_Yeru: WORD 0xD0AB
UxCapital_SoftSign: WORD 0xD0AC
UxCapital_E: WORD 0xD0AD
UxCapital_Yu: WORD 0xD0AE
```

```
UxCapital_Ya: WORD 0xD0AF
UxSmall_A: WORD 0xD0B0
UxSmall_Be: WORD 0xD0B1
UxSmall_Ve: WORD 0xD0B2
UxSmall_Ghe: WORD 0xD0B3
UxSmall_De: WORD 0xD0B4
UxSmall_Ie: WORD 0xD0B5
UxSmall_Zhe: WORD 0xD0B6
UxSmall_Ze: WORD 0xD0B7
UxSmall_I: WORD 0xD0B8
UxSmall_ShortI: WORD 0xD0B9
UxSmall_Ka: WORD 0xD0BA
UxSmall_El: WORD 0xD0BB
UxSmall_Em: WORD 0xD0BC
UxSmall_En: WORD 0xD0BD
UxSmall_O: WORD 0xD0BE
UxSmall_Pe: WORD 0xD0BF
UxSmall_Er: WORD 0xD180
UxSmall_Es: WORD 0xD181
UxSmall_Te: WORD 0xD182
UxSmall_U: WORD 0xD183
UxSmall_Ef: WORD 0xD184
UxSmall_Ha: WORD 0xD185
UxSmall_Tse: WORD 0xD186
UxSmall_Che: WORD 0xD187
UxSmall_Sha: WORD 0xD188
UxSmall_Shcha: WORD 0xD189
UxSmall_HardSign: WORD 0xD18A
UxSmall_Yeru: WORD 0xD18B
UxSmall_SoftSign: WORD 0xD18C
UxSmall_E: WORD 0xD18D
UxSmall_Yu: WORD 0xD18E
UxSmall_Ya: WORD 0xD18F // КОНЕЦ МАССИВА ЭЛЕМЕНТОВ UTF-8
```