

Текст задания:

Разработать программу, заменяющую все заглавные буквы в заданной ASCII-строке прописными.

Код программы:

```
format PE console
include 'win32a.inc'
entry start
```

```
section 'data' data readable writeable
```

```
    EnterString db 'Write the string',10,0 ;Строки для вывода 10 -
    модификатор перехода на новую строку, 0 - модификатор конца строки
    tempString db '%s', 0 ;Для ввода строк %s - значит туда могут
    передаваться строки
```

```
    A db 256 dup(?) ; массив для хранения букв введенной строки
```

```
    B db 256 dup(?) ; массив для хранения букв с обработанной строки
```

```
section 'text' code executable readable
```

```
start:
```

```
    push StringOutOne ; передача в стек параметров
```

```
    call [printf] ; некоторые функции берут параметры из стека по
    принципу pop
```

```
    push A
```

```
    push strScanString
```

```
    call [scanf] ; вызов функции считывания строки и записывания ее в
    массив
```

```
    push B
```

```
    push A
```

```
    call toLower ; вызов метода аналогичного toLower
```

```
    push B
```

```
    call [printf] ; вывод результата
```

```
    call [getch] ; ожидание нажатия клавиши
```

```
    push 0
```

```
    call [ExitProcess]
```

proc toLower inputLine, outputLine ;объявляю название функции и ее переменную

mov edi,[inputLine] ; передаю в edi ссылку на первый байт массива inputLine
(inputLine это переменная в которой лежит A)

mov ecx,256 ; итератор для цикла rep (макс длина строки). rep работает за счет того что ecx уменьшается на 1 при каждой итерации цикла

xor al,al ; заданию нулевое значение регистра al с которым работает scasb для определения конца строки (сравнивает текущий элемент в edi с al)
repnz scasb ; repz - повторяет цикл пока Zflag = true, repnz - пока Zflag =false

sub edi,[inputLine] ; inputLine указывает на позицию начала строки. edi уже указывает на конец строки. отнимаю от edi inputLine чтобы узнать количество символов
sub edi,1 ; так надо

mov ecx,edi ; теперь мы точно знаем размер строки

mov eax,[inputLine] ; регистр на начало inputLine
mov ebx,[outputLine] ; регистр на начало outputLine

.loop:
mov dl,[eax] ; al отвечает за младший байт в eax поэтому если изменю его, то изменю и eax

cmp dl,'A' ; сравниваю значение dl с значением чара inputLine
jb .skip ; jl для signed переменных, jb для unsigned переменных (проверка на флаг сравнения). Если проходит условие, то переходим в .skip

cmp dl,'Z'
ja .skip ; ja unsigned , jg signed (больше)
;если эти условия не прошли, значит надо уменьшить заглавную букву
add dl,'a'-'A' ; добавляю сколько надо чтобы получить нужный знак из таблицы ASCII

.skip:
mov [ebx], dl ; вставляю значение inputLine через dl в outputLine
add eax,1
inc ebx ;разные способы увеличить ~~ (sub eax,-1) ы

loop .loop ; команда loop уменьшает ecx на 1 и если он не равен 0 то переходит в .loop

ret
endp ; закрытие макроса proc

```

section '.idata' import data readable
library kernel, 'kernel32.dll',\
    msvcrt, 'msvcrt.dll',\
    user32, 'USER32.DLL'

include 'api\user32.inc'
include 'api\kernel32.inc'
import kernel,\
    ExitProcess, 'ExitProcess'

include 'api\kernel32.inc'
import msvcrt,\
    printf, 'printf',\
    scanf, 'scanf',\
    getch, '_getch'

```

В данной программе я получаю строку на ввод, преобразую ее в массив чисел (назовем это массив char).

Узнаю, какого он размера

Затем провожу операцию изменения чисел в элементах массива (в соответствии с необходимыми значениями из таблицы ASCII, проверяя каждый элемент массива)

Все числа соответствующие диапазону char-ов от А до Z преобразуются в их строчные варианты (кириллический алфавит не затрагивался)

Вывожу получившийся результат.

```
Write the string
thereisnobigletter
thereisnobigletter
```

```
Write the string
fpsaogiireAOKFjsjfdpkskfpewfkaPO
fpsaogiireaokfjsjfdpkskfpewfkapo_
```

```
Write the string
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
```

```
Write the string
33u3gj90]qFDSF<rrj934g-щалщоп9ШЩоаы
33u3gj90]qfdsf<rrj934g-щалщоп9ШЩоаы_
```

При введении строк через пробел будет учтено только первое слово.