

Elektrotehnički fakultet Univerziteta u Beogradu

Projekat iz Računarske elektronike

Autori:

Luka Jevremović 0662/2013

Tatjana Toroman 0282/2012

Jun, 2016.

Kod projekta

```
INCLUDE Irvine32.inc

INCLUDE Macros.inc
ExitProcess PROTO, dwExitCode:dword

.data
tabela WORD 'P', 'L', 'A', 'Y', 'F', 'I', 'R', 'B', 'C', 'D', 'E', 'G', 'H', 'K', 'M',
'N', 'O', 'Q', 'S', 'T', 'U', 'V', 'W', 'X', 'Z', 'p', 'l'
        WORD 'a', 'y', 'f', 'i', 'r', 'b', 'c', 'd', 'e', 'g', 'h', 'k', 'm', 'n', 'o',
'q', 's', 't', 'u', 'v', 'w', 'x', 'z'

cHandle HANDLE 0

.data?
ch1 WORD ?
ch2 WORD ?
.code

citaj proc

call ReadChar
cmp al, 'j'
jne next1
mov al, 'i'

next1 : mov ah, 0
        mov ch1, ax
        call ReadChar
        cmp al, 'j'
        jne next2
        mov al, 'i'
        mov ah, 0; poistovecuje i i j

next2: mov ah, 0
        mov ch2, ax

ret
citaj endp

zamjeni proc uses eax ecx
LOCAL mask1 : WORD, mask2 : WORD, i1 : WORD, i2 : WORD, j1 : WORD, j2 : WORD, skip:WORD
mov ecx, 0
mov mask1, 0
mov mask2, 0
mov skip, 0
loop1 : mov ax, tabela[ecx*2]
        inc ecx
        cmp ax, ch1
        jnz loop1
        dec ecx
        mov i1, cx
        mov ecx, 0 ; pronalazi prvi karakter u tabeli
        cmp i1, 49
        jc loop2
        mov skip, 1; preskace kodovanje ako znak nije slovo
```

```

loop2 : mov ax, tabela[ecx * 2]
        inc ecx
        cmp ax, ch2
        jnz loop2
        dec ecx
        mov i2, cx ;pronalazi drugi karakter u nizu
        cmp i2, 49
        jc skip1
        mov skip,1
skip1: cmp skip, 1
        jz preskok

        cmp i1, 25
        jc ok1
        sub i1,25
        mov mask1,1
ok1:   cmp i2, 25
        jc ok2
        sub i2, 25
        mov mask2, 1 ;pretvaranje indexa u slucaju malih slova

ok2:   mov cx, 5
        mov ax, i1
        sub edx,edx
        div cx
        mov j1, ax
        mul cx
        sub i1, ax
        mov ax, i2
        sub edx, edx
        div cx
        mov j2, ax
        mul cx
        sub i2, ax ; pretvara indexe niza u indexe matrice

        mov ax, j1 ;kodovanje indexa
        cmp ax, j2
        jne next1
        inc i1
        inc i2
        jmp kraj
next1 : mov ax, i1
        cmp ax, i2
        jne next2
        inc j1
        inc j2
        jmp kraj
next2 : mov ax, i1
        xchg ax, i2
        mov i1, ax

        kraj : cmp i1, 5
                jne next3
                mov i1, 0
next3:  cmp i1, -1
                jne next4
                mov i1, 4

```

```

next4 : cmp i2, 5
        jne next5
        mov i2, 0
next5 : cmp i2, -1
        jne next6
        mov i2, 4

next6 : cmp j1, 5
        jne next7
        mov j1, 0
next7 : cmp j1, -1
        jne next8
        mov j1, 4
next8 : cmp j2, 5
        jne next9
        mov j2, 0
next9 : cmp j2, -1
        jne next10
        mov j2, 4 ;vraca indexe matrice u indexe niza

next10:      mov ax,j1
             mov cx,5
             mul cx
             add i1,ax
             mov ax,j2
             mul cx
             add i2,ax

             cmp mask1,0
             jz ok3
             add i1,25 ;mask1 i  sluzi za razlikovanje malih i velikih slova
ok3:      cmp mask2,0
             jz ok4
             add i2, 25;vraca indexe matrice u indexe niza

ok4:

             mov cx,i1
             mov ax, tabela[ecx*2]
             mov ch1,ax
             mov cx, i2
             mov ax, tabela[ecx * 2]
             mov ch2,ax
preskok:

ret
zameni endp

pisi proc uses eax
LOCAL bytesWritten : DWORD

INVOKE GetStdHandle, STD_OUTPUT_HANDLE
mov cHandle ,eax
INVOKE WriteConsole, cHandle, ADDR ch1, 1, ADDR bytesWritten,0
INVOKE WriteConsole, cHandle, ADDR ch2, 1, ADDR bytesWritten,0
ret

```

```
pisi endp
```

```
main proc  
again:  
call citaj  
call zameni  
call pisi  
jmp again
```

```
invoke ExitProcess,0  
main endp  
end main
```

Funkcija: čitaj

Funkcija čitaj služi da pročita karaktere sa ulaza. Pošto je slova "i" i "j" poistovećuju, proverava se da li su baš ta slova učitana i vrši se zamena.

Funkcija: zameni

Funkcija zameni služi da zameni slova po principu Playfair-ovog šifrovanja zadatog u postavci. Promenljive mask1 i mask2 služe da razlikuju velika slova od malih.

Skip služi da, ukoliko pročitani karakter nije slovo, preskoči taj karakter dok ne dođe do narednog slova.

Indeksi i1 i i2 se upoređuju sa 25 da bismo odredili da li je karakter malo ili veliko slovo. Ako je u pitanju malo slovo, oduzimamo 25 i pretvaramo ga u veliko i obrada se radi istim algoritmom kao i za velika slova. Kasnije se veliko slovo ponovo vraća u malo, ukoliko je na početku bilo malo, što se pamti u promenljivima mask1 i mask2.

Funkcija: piši

Funkcija piši služi da ispiše dobijene karaktere na izlaz.

Napomena: Svaka petlja u kodu ima posebno ime jer kompajler nije hteo da prihvati anonimne libele.