

# Projekat iz računarske elektronike

Dragana Jevtić 685/2013  
Andjelija Ponjavić 39/2013  
odsek OE

#### 4. projekat

Napisati program koji sa standardnog ulaza čita broj  $n$  koji predstavlja pomeraj prilikom primene Cezarovog kodiranja. Napisati funkciju koja po unosu pomeraja mapira tipke tastature tako da se prilikom kucanja poruke na standardnom izlazu prikazuje šifrovana poruka. Cezarovo šifrovanje preslikava slovo u slovo pomerenom  $n$  mesta.

Na primer za  $n = 3 : A \rightarrow D, B \rightarrow E, C \rightarrow F$ .

#### Izveštaj

Po pokretanju programa u consoli se ispisuje prouka koja od korisnika traži da unese pomeraj  $n$ . Podatak se čita pomoću funkcije *ReadInt* a potom smešta u promenljivu  $n$ . Zatim se vrši provera ispravnosti unesene vrednosti u sekciji *check*. Ukoliko je unesen negativan broj prelazi se na sekciju *err* u kojoj se ispisuje poruka na izlazu i završava program. Ukoliko je uneseni pomeraj validan, vrši se njegova korekcija, oduzimanjem 26 od  $n$  u petlji, do zadovoljavanja uslova da se  $n$  nalazi u osnovnom opsegu od 0 do 25. Zatim se prelazi na deo obrade unesenih karaktera u sekciji *read*. Korišćenjem funkcije *ReadKey* čita se jedan uneseni karakter i potom proverava da li je u pitanju slovo. Ukoliko ukucani karakter nije slovo prelazi se u sekciju *stop* i izlazi iz programa. U suprotnom, program se grana na dva podslučaja *con1* i *con2*, prema tome da li je ukucano veliko ili malo slovo. Zatim se vrši konverzija slova prema Cezarovom algoritmu, tako što se na ascii kod unesenog karaktera dodaje pomeraj, ukoliko ova vrednost izlazi iz opsega odgovarajućih slova, oduzima se 26 i dobijena vrednost predstavlja ascii kod pomerenog slova. Potom se prelazi na ispis u sekciji *write*. Korišćenjem *handle*-a i pozivom funkcije *WriteConsole* ipisuje se dobijeni *char* na izlaz. Program se zatim vraća na početak i obradjuje slovo po slovo sve dok se ne unese neki karakter koji nije slovo, nakon čega se napušta program.

#### Kod

```

1  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2
3  INCLUDE Irvine32.inc
4  INCLUDE Macros.inc
5
6  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
7
8  .data
9  message BYTE "Unesite pomeraj, n=", 0
10 messageSize DWORD ($-message)
11
12 error BYTE "Greska! Neispravan unos!", 0
13 errorSize DWORD ($-error)
14
15 key BYTE ?
16 keySize WORD ($-key)
17
18 consoleHandle HANDLE 0
19
20 bytesWritten BYTE ?
21
22 n DWORD ?
23
24 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
25
26 .code
27 main PROC
28     ; Get n from the console using ReadInt
29     mov edx, OFFSET message
30     call WriteString    ; display string
31     call ReadInt       ; read integer into eax
32     mov esi, eax       ; store in esi
33     mov n, esi         ; store in n
34
35 check: ; Check if the n is less than 0 or bigger than 26
36     cmp n, 0
37     jl err
38     cmp n, 26
39     jb start
40     mov eax, n
41     sub eax, 26
42     mov n, eax
43     jmp check
44
45 start: ; Start
46     ; Get the console output handle
47     INVOKE GetStdHandle, STD_OUTPUT_HANDLE
48     mov consoleHandle, eax
49
50 read: ; Read the key
51     call ReadKey ; wait for a keypress
52     jz read
53     mov n, esi
54     cmp al, 65
55     jb stop
56     cmp al, 90
57     jbe con2
58     cmp al, 97
59     jb stop
60     cmp al, 122
61     ja stop
62
63 con1: ; Convert the lower-case
64     add al, BYTE PTR n
65     cmp al, 122
66     jbe write
67     sub al, 26
68     jmp write
69
70 con2: ; Convert the upper-case
71     add al, BYTE PTR n
72     cmp al, 90
73     jbe write
74     sub al, 26

```

```

75
76 write: ; Display replaced character
77     mov key, al
78     INVOKE WriteConsole,
79         consoleHandle,
80         ADDR key,
81         keySize,
82         ADDR bytesWritten,
83         0
84     jmp read
85
86 err: ; Display the error message
87     ; Get the console output handle
88     INVOKE GetStdHandle, STD_OUTPUT_HANDLE
89     mov consoleHandle, eax
90
91     INVOKE WriteConsole,
92         consoleHandle,
93         ADDR error,
94         errorSize,
95         ADDR bytesWritten,
96         0
97
98 stop: ; Exit program
99     INVOKE ExitProcess, 0
100
101 main ENDP
102
103 END main

```