;Contorizarea consoanelor dintr-un text introdus de la tastatura

;Declarare segmente de lucru

DOSSEG

.model small ; modul de utilizare al memoriei

.stack ; alocarea segmentului de stiva la valoarea implicita 256 octeti

.data ; sectiune date

mesaj1 db 'Introduceti sirul: $'

mesaj2 db 13,10,'Numarul consoanelor este: $'

sir db 100,10 dup(?) ; container - max 100 caractere terminate cu LF

.code ;inceputul segmentului de cod

start: ;eticheta de inceput

mov ax,@data ;DS se incarca cu adresa segmentului de date

mov ds,ax

mov ah,09h ; cod functie de afisare a unui sir de caractere

mov dx,offset mesaj1 ; de la adresa specificata prin variabila mesaj1

int 21h ; intrerupere cu functie de afisare caracter

mov ah,0ah ; int 21/AH=0ah - citeste string de la stdin

mov dx,offset sir ;intr-o zona tampon avand adresa de inceput sir+2

int 21h

mov si,offset sir

xor cl,cl ; cx low e initializat cu zero, aici va contoriza

analiza:

mov dl,[si+2] ;dx low primeste succesiv cate un caracter

cmp dl,13 ;se verifica daca este sfarsitul sirului (CR)

je iesire

cmp dl,'b' ; compara cu fiecare consoana pe rand si o numara daca a gasit-o

je contor

cmp dl,'c'

je contor

cmp dl,'d'

je contor

cmp dl,'f'

je contor

cmp dl,'g'

je contor

cmp dl,'h'

je contor

cmp dl,'j'

je contor

cmp dl,'k'

je contor

cmp dl,'l'

je contor

cmp dl,'m'

je contor

cmp dl,'n'

je contor

cmp dl,'p'

je contor

cmp dl,'q'

je contor

cmp dl,'r'

je contor

cmp dl,'s'

je contor

cmp dl,'t'

je contor

cmp dl,'v'

je contor

cmp dl,'w'

je contor

cmp dl,'x'

je contor

cmp dl,'y'

je contor

cmp dl,'z'

je contor

continua: ; ajunge aici implicit cand nu e consoana sau explicit cand a gasit si numarat

inc si ; litera urmatoare

jmp analiza

contor:

inc cl ; mai contorizeaza una vocala

jmp continua ;

iesire:

mov ah,09h ; Int 21/AH=09h - scrie string pe stdout

mov dx,offset mesaj2 ; adresa de unde ia mesajul terminat cu $( DS:DX)

int 21h

mov dl,cl

add dl,30h ; conversie in ASCII 48+i

mov ah,02h

int 21h

mov ah,4ch ; EXIT - TERMINATE WITH RETURN CODE

int 21h

end start

DOSSEG

.model small

.stack

.data

mesaj1 db 'Introduceti sirul: $'

mesaj2 db 13,10,'Numarul consoanelor este: $'

sir db 100,10 dup(?)

.code

start:

mov ax,@data

mov ds,ax

mov ah,09h

mov dx,offset mesaj1

int 21h

mov ah,0ah

mov dx,offset sir

int 21h

mov si,offset sir

xor cl,cl

analiza:

mov dl,[si+2]

cmp dl,13

je iesire

cmp dl,'b'

je contor

cmp dl,'c'

je contor

cmp dl,'d'

je contor

cmp dl,'f'

je contor

cmp dl,'g'

je contor

cmp dl,'h'

je contor

cmp dl,'j'

je contor

cmp dl,'k'

je contor

cmp dl,'l'

je contor

cmp dl,'m'

je contor

cmp dl,'n'

je contor

cmp dl,'p'

je contor

cmp dl,'q'

je contor

cmp dl,'r'

je contor

cmp dl,'s'

je contor

cmp dl,'t'

je contor

cmp dl,'v'

je contor

cmp dl,'w'

je contor

cmp dl,'x'

je contor

cmp dl,'y'

je contor

cmp dl,'z'

je contor

continua:

inc si

jmp analiza

contor:

inc cl

jmp continua

iesire:

mov ah,09h

mov dx,offset mesaj2

int 21h

mov dl,cl

add dl,30h

mov ah,02h

int 21h

mov ah,4ch

int 21h

end start