;Numarare vocale dintr-un text introdus de la tastatura

;Declararea 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 vocalelor este: $'

sir db 100,10 dup(?) ; container - max 1a00 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,'a' ; compara cu fiecare vocala pe rand si o numara daca a gasit-o

je contor

cmp dl,'e'

je contor

cmp dl,'i'

je contor

cmp dl,'o'

je contor

cmp dl,'u'

je contor

cmp dl,'?'

je contor

cmp dl,'?'

je contor

continua: ; ajunge aici implicit cand nu e vocala 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