org 100h

init: lea bx, inputString // Incarca adresa efectiva a sirului inputString in BX-registru special conceput pentru a stoca adrese efective.

mov si, 0h //initializam SI-sorce index-variabila care parcurge practic sirul inputString, cu 0h

mov cx, charX-inputString // facem diferenta intre adresa efectiva a caracterului charX si sirul inputString, si aflam practic numarul de caractere al sirului inputString. Numarul de caractere al sirului va fi mutat in contorul CX

mov dx, 0h //dx reprezinta registrul care stocheaza cate inlocuiri vor fi facute.Acesta este initializat cu 0h

mov ah, [charY] // mutam in byte-ul cel mai semnificant al registrului AX , informatia continuta de char(codul ASCII)

compare: mov al,[bx+si] //muta valoarea stocata la adresa efectiva bx +si in registrul p 8 biti al

cmp al,[charX] // compar valoarea din al cu valoarea stocata in charX. Valorile sunt comparate prin scadere.instructiunea nu modifica valoarea lui al. Steagul ZF se modifica aici. Daca cele doua numere sunt egale,ZF este 1, daca nu este 0.

je replace //daca valoarea din al, adica valoarea curenta din sir este egala cu valoarea din charX , programul sare la eticheta replace, daca nu, programul trece mai departe.

nextElem:inc si // daca am ajuns aici, inseamna ca valoarea din al, din sir nu a fost egala cu valoarea din charX, de unde rezulta ca trebuie sa parcurgem in continuare sirul.Incrementam registrul SI, pentru a merge la valoarea urmatoare din sir

loop compare // loop la eticheta compare.facem asta pana cand sirul se termina. Aici se decrementeaza CX. Cand CX este 0, inseamna ca sirul s-a terminat.

int 20h // directiva de asamblare. Programul s-a terminat

replace: mov [bx+si],ah // aici ajungem daca cele 2 elemente comparate,al si valoarea din charX sunt egale. Daca sunt egale, valoarea din sir, trebuie inlolcuita cu valoarea lui charY. Instructiunea asta, muta la adresa curenta din stiva, adresa bx +si, care este si adresa efectiva a elementului care trebuie inlocuit, valoarea ah, valoarea elementului charY, cel cu carese inlocuieste

inc dx//am inlocuit un element. Contorul dx creste

jmp nextElem //sarim la urmatorul element. Se incrementeaza si pentru a ajunge la uratoarea valoare din sir. Intram iar in compare, conform loop

inputString db 'felix o sa aiba zece la sda' //initializarea sirului de caracere

charX db 'e' //initializarea caracterului care va fi inlocuit

charY db 'm' //initializarea caracterului cu care va fi inlocuit charX

///{Elementele sirului sunt pe un byte, adica pe 8-biti si cele 2 variabile charX si charY sunt tot pe 8 biti. Din cauza asta, ele sunt definite folosind db, directiva de asamblare define byte. Variabilele sunt plasate in memorie la o adresa efectiva }. {Elementele de tip caractere sunt reprezentate in memorie ca coduri ASCII}.