

bits 32

global start

extern exit, printf, scanf, fopen, fread, fclose

import exit msvcrt.dll

import printf msvcrt.dll

import scanf msvcrt.dll

import fopen msvcrt.dll

import fread msvcrt.dll

import fclose msvcrt.dll

segment data use32 class=data

 ; ...

 n dd -1

 index db 0

 file_name resb 100

 access_mode db "r", 0

 format_afisare1 db "nume fisier: ", 0

 format_afisare2 db "n: ", 0

 format_citire_fisier db "%s", 0

 format_citire_numar db "%d", 0

 format_afisare_cuvant db "%s", 10, 0

 cuvant resb 100

 file_descriptor dd -1

segment code use32 class=code

 start:

 ; ...

 ; citim datele fisierul si numarul de la tastatura

push format_afisare1

call [printf]

add esp, 4 * 1

push file_name

push format_citire_fisier

call [scanf]

add esp, 4 * 2

push format_afisare2

call [printf]

add esp, 4 * 1

push n

push format_citire_numar

call [scanf]

add esp, 4 * 2

; -----

;deschidem fisierul

push dword access_mode

push dword file_name

call [fopen]

add esp, 4 * 2

mov [file_descriptor], eax

cmp eax, 0 ; inchidem programul daca nu a fost deschis corect fisierul

je final

; cu esi mergem mai departe pe sir in timp ce cuvnt preia cuvntul actual pentru al afisa daca este cazul

repet2:

mov esi, cuvnt

; citim litera cu litera cuvntul

repet1:

; citim un caracter

push dword [file_descriptor]

push dword 1

push dword 1

push dword esi

call [fread]

add esp, 4*4

inc esi

;daca jungem la final de fisier

cmp eax, 0

je final_repet1

; daca am ajuns la finalul cuvntului (adica la spatiu)

; iesim din repetitiva pentru a verifica cuvntul si al catelea e

cmp byte [esi-1], 32

je final_repet1

jmp repet1

final_repet1:

cmp byte [cuvnt], 0 ; verificam daca cuvntul e gol

je final_repet2

```

mov byte [esi - 1], 0; cum citim si spatiul, trebuie sa scapam de el
;verificam daca cuvantul actual este pe o pozitie multipla de numarul introdus de la tastatura
mov eax, 0
mov edx, 0
mov al, [index]
mov ebx, [n]
div bx
cmp dx, 0
je skipPrint
;daca nu e atunci sarim peste print si mergem mai departe cu citirea
push cuvant
push format_afisare_cuvant
call [printf]
add esp, 4 * 1

skipPrint:
mov byte [cuvant], 0
inc byte [index]
jmp repeta2
final_repeta2:

;inchidem fisierul
push dword [file_descriptor]
call [fclose]
add esp, 4
push dword 0
final:
call [exit]

```