Text

Description automatically generated

A picture containing text

Description automatically generatedGraphical user interface, text, application

Description automatically generated

bits 32 ; assembling for the 32 bits architecture

; declare the EntryPoint (a label defining the very first instruction of the program)

global start

; declare external functions needed by our program

extern exit,fopen,fprintf,fclose

import exit msvcrt.dll ; exit is a function that ends the calling process.

import fopen msvcrt.dll ; msvcrt.dll contains exit, printf

import fprintf msvcrt.dll

import fclose msvcrt.dll

; our data is declared here (the variables needed by our program)

segment data use32 class=data

nume\_fisier db "output.txt",0

acces\_mode db "w",0

dif db 0

description\_fis dd -1

text db "Tata are 5/\*-2 feciori",0

l equ $-text-1

nw\_text times l+1 db 0

; our code starts here

segment code use32 class=code

start:

;pentru modificarea textului vom folosi operatii pe siruri

cld ;parcurgem in de la stanga la dreapta

mov ESI,text ;in SOURCE INDEX punem adresa textului initial

mov EDI,nw\_text ; in DESTINATION INDEX punem adresa textului de construit

mov ECX,l ;pune in ECX lungimea sirului

mov byte[dif],'A'-'a';

st\_loop:

lodsb ;luam in AL caracterul din text

;verificam if('a'<=c && c<='z')

cmp AL,'z'

ja et1

cmp AL,'a'

jb et2

add AL,[dif] ;in caz afirmativ adaugam cont

et1:

et2:

stosb ;pune in sirul destinatie pe AL

loop st\_loop

mov AL,0 ; adaugam 0-ul final pentru a putea afisa in fisier

stosb

;apelam fopen pentru a deschide fisierul

;EAX=fopen(nume\_fisier,acces\_mode)

push dword acces\_mode

push dword nume\_fisier

call [fopen]

add ESP,4\*2

mov [description\_fis],EAX ;salvam valoare returnata in EAX in description\_fis

cmp EAX,0

je final

;scriem testul in fisier

;fprintf(description\_fis,nw\_text)

push dword nw\_text

push dword [description\_fis]

call [fprintf]

add ESP,4\*2

;inchidem fisierul

;fclose(description\_fis)

push dword [description\_fis]

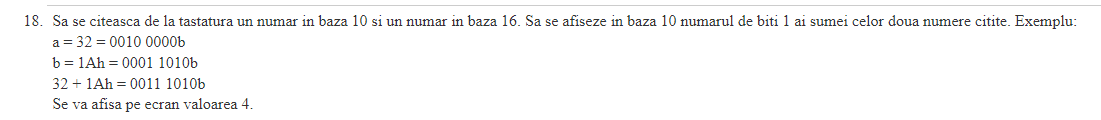
call [fclose]

add ESP,4

final:

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program

Graphical user interface

Description automatically generatedGraphical user interface, text, application

Description automatically generated

mov [description\_fis],EAX ;salvam valoare returnata in EAX in description\_fis

cmp EAX,0

je final

;scriem testul in fisier

;fprintf(description\_fis,nw\_text)

push dword nw\_text

push dword [description\_fis]

call [fprintf]

add ESP,4\*2

;inchidem fisierul

;fclose(description\_fis)

push dword [description\_fis]

call [fclose]

add ESP,4

final:

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program

bits 32

global start

extern exit,scanf,printf

import exit msvcrt.dll ; exit is a function that ends the calling process.

import scanf msvcrt.dll ; msvcrt.dll contains exit, printf

import printf msvcrt.dll

; our data is declared here (the variables needed by our program)

segment data use32 class=data

a dd 0

b dd 0

cont dd 0

format\_dec dd "%d",0

format\_hex dd "%x",0

; our code starts here

segment code use32 class=code

start:

;scanf("%d",&a)

push dword a

push dword format\_dec

call [scanf]

add ESP,4\*2

;scanf("%x",&b)

push dword b

push dword format\_hex

call [scanf]

add ESP,4\*2

mov EAX,[a]

add EAX,[b]

mov ECX,0 ;in ECX se numara bitii de 1

;numaram biti de 1 ai rezultatului

cnt:

mov dword[a],1

and [a],EAX

add ECX,[a]

shr EAX,1

jnz cnt

mov [cont],ECX ;punem in cont ce vrem sa afisam

;printf("%d",cont)

push dword [cont]

push dword format\_dec

call [printf]

add ESP,4\*2

push dword 0 ; push the parameter for exit onto the stack

call [exit] ; call exit to terminate the program