**Proiect ASM**

1. **Descrierea programului**

Programul este conceput astfel incat sa afiseze daca un numar introdus de la tastatura este par, impar sau egal cu zero.

1. **Codul sursa**

.model small

.stack 100h

.data

nnegativ db 0ah, 0dh, 'negativ$'

ppozitiv db 0ah, 0dh, 'pozitiv$'

zzero db 0ah, 0dh, 'zero$'

.code

start:

mov ax, @data

mov ds, ax

mov ah, 1

int 21h

mov bl, al

mov cl, 30h

cmp bl, cl

jl negativ

je zero

jg pozitiv

negativ:

mov dx, offset nnegativ

mov ah, 09h

int 21h

jmp END

zero:

mov dx, offset zzero

mov ah, 09h

int 21h

jmp END

pozitiv:

mov dx, offset ppozitiv

mov ah, 09h

int 21h

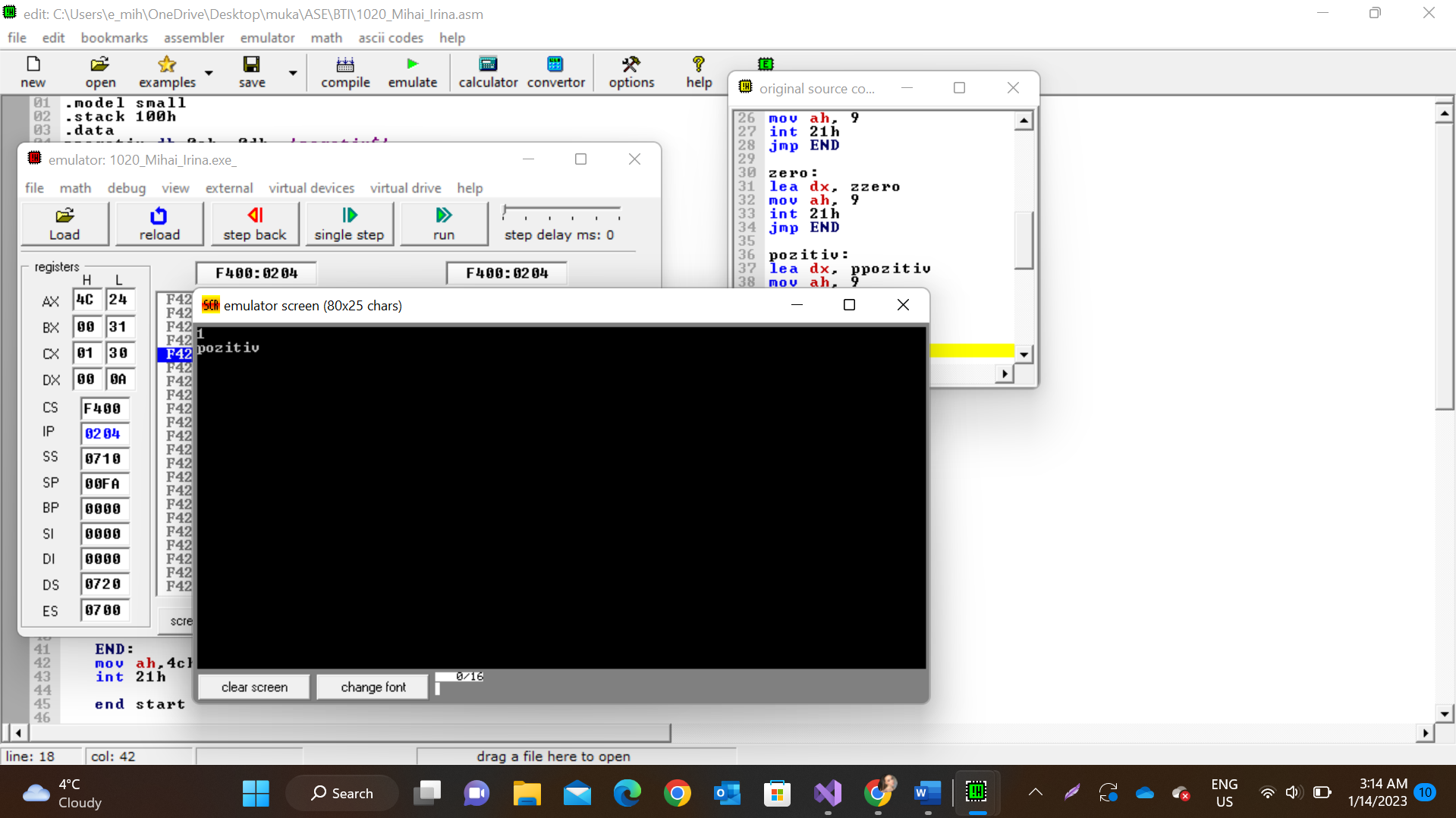
END:

mov ah,4ch

int 21h

end start

1. **Rezultate program**



A screenshot of a computer

Description automatically generated with medium confidence

1. **Explicatii**

.model small

.stack 100h – dimensiunea stivei de 100h octeti

.data – defineste variabilele ce vor fi folosite in codul sursa ( sunt definite 3 variabile : nnegativ, ppoziti, zzero)

nnegativ db 0ah, 0dh, 'negativ$'

ppozitiv db 0ah, 0dh, 'pozitiv$'

zzero db 0ah, 0dh, 'zero$'

.code – reprezinta unde se afla codul sursa ( intrarea in cod este eticheta ‘start’)

start:

mov ax, @data – se muta o valoare din memorie de la adresa ‘@data’ in registrul ax

mov ds, ax - se muta continutul registrului ax in registrul ds

mov ah, 1 – setam registrul ah la 1

int 21h – se ruleaza codul si introducem un numar de la tastatura

mov bl, al – numarul citit de la tastatura se pune in registrul al

mov cl, 30h – setam registrul cl la 30h

cmp bl, cl - compara cele doua numere

jl negativ - daca bl<cl afisez ‘negativ’

je zero – daca bl=cl afisez ‘zero’

jg pozitiv – daca bl>cl afisez ‘pozitiv’

negativ:

mov dx, offset nnegativ – se aloca adresa memoriei nnegativ in registrul dx

mov ah, 09h – se muta valoarea 9 in registrul ah => urmatoarea instructiune int 21h va afisa un mesaj

int 21h – se ruleaza programul si afiseaza mesajul ‘negativ’ de pe stiva

jmp END – se sare la eticheta END

zero:

mov dx, offset zzero – se preia adresa memoriei variabilei zzeo si se stocheaza in registrul dx

mov ah, 09h – se stocheaza val 9 in registrul ah

int 21h – se apeleaza un serviciu al sistemului de operare care afiseaza mesajul stocat in registrul ah

jmp END – se sare la eticheta END

pozitiv:

mov dx, offset ppozitiv – se stocheaza adresa de inceput a sirului de caractere in registrul dx

mov ah, 09h – se stocheaza val 9 in registrul ah

int 21h - se apeleaza un serviciu al sistemului de operare care afiseaza mesajul stocat in registrul ah

END:

mov ah,4ch – se seteaza registrul ah la 4ch

int 21h – se ruleaza programul si se iese din acesta

end start – sfarsitul programului