

ACADEMIA DE STUDII ECONOMICE, BUCUREȘTI

Facultatea de Cibernetică, Statistică și Informatică Economică

Proiect BTI

STUDENT: MATEI TUDOR-MIHAI

SPECIALIZAREA: INFORMATICA ECONOMICA

AN DE STUDIU: I

GRUPA: 1020

SERIA: D

AN UNIVERSITAR: 2022-2023

Descrierea programului

Programul verifica daca un caracter introdus de utilizator este litera mare sau mica. Se foloseste instructiunea INT 21H pentru a primi inputul de la utilizator. Se foloseste apoi instructiunea CMP pentru a compara inputul cu valorile "A" si "Z" pentru a vedea daca este litera mare. Daca da, PRINT afiseaza mesajul "YOUR CASE : IS UPPER CASE!". Apoi, se verifica daca este litera mica, comparandu-l cu "a" si "z". Daca da, PRINT afiseaza mesajul "YOUR CASE : IS LOWER CASE!". Daca inputul nu este litera mare sau mica, PRINT afiseaza mesajul "NONE OF THEM!". La sfarsit, se foloseste instructiunea JMP EXIT pentru a sari peste instructiunile urmatoare.

Cod Sursa:

```
INCLUDE "EMU8086.INC"

.MODEL SMALL

.STACK 100H

.CODE

MAIN PROC

    PRINT "INPUT A CHARACTER:  "

    MOV AH, 1
    INT 21H
    MOV BL, AL
```

```
CMP BL, "A"
JNGE NONE
CMP BL, "Z"
JNLE LOW
PRINTN
PRINT "YOUR CASE : IS UPPER CASE!"
JMP EXIT
```

LOW:

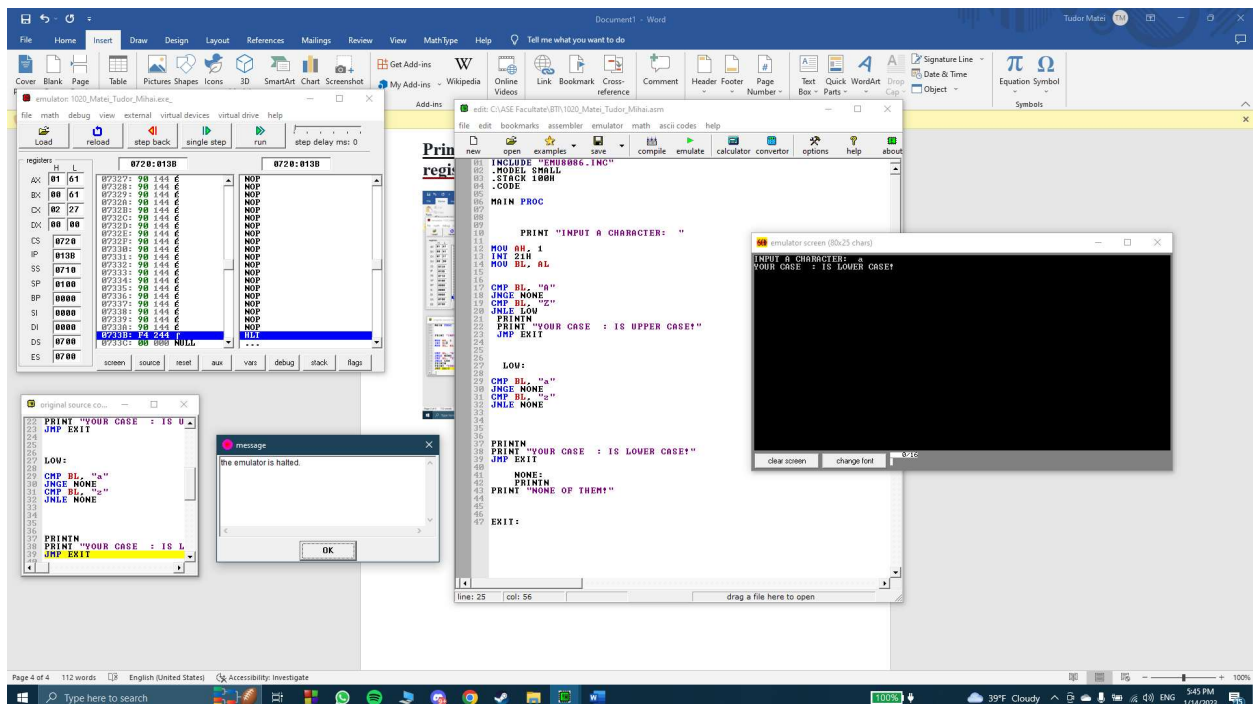
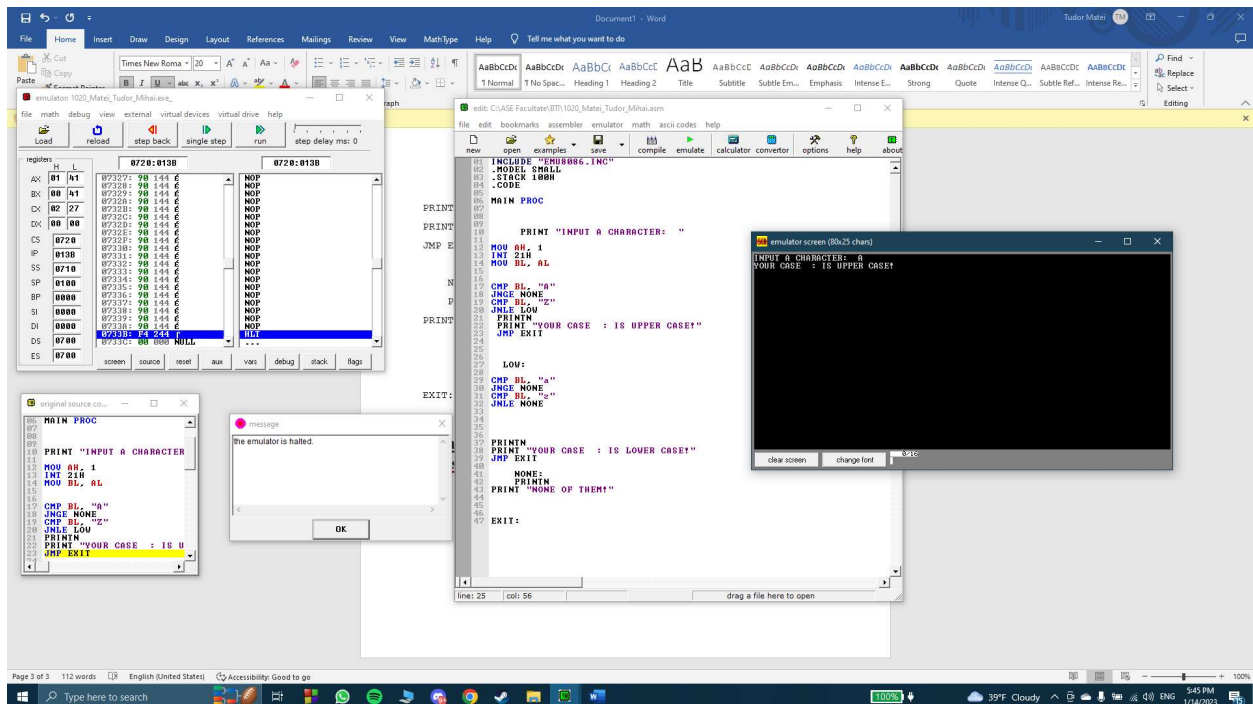
```
CMP BL, "a"
JNGE NONE
CMP BL, "z"
JNLE NONE
```

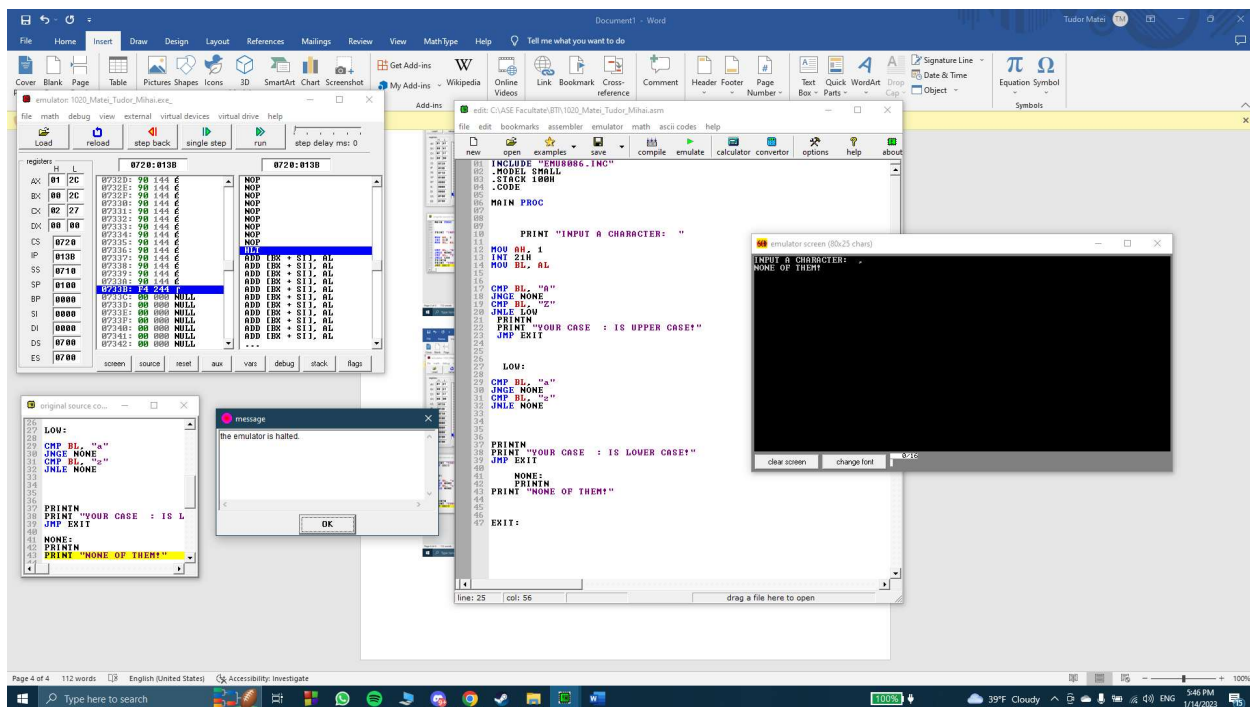
```
PRINTN
PRINT "YOUR CASE : IS LOWER CASE!"
JMP EXIT
```

NONE:

```
PRINTN
PRINT "NONE OF THEM!"
EXIT:
```

Print screen cu rezultatele programului in memorie si registrii





Explicatii

INCLUDE "EMU8086.INC"	Aceasta functie include un set de macro-uri si instructiuni. Aceste macro-uri si instructiuni ofera scurtaturi utile pentru lucrul cu registri, gestionarea zonelor de memorie si alte elemente specifice procesorilor de 16 biti.
.MODEL SMALL	Este o directivă de compilare care este utilizată pentru a specifica modelul de cod folosit pentru compilarea codului sursă.
.STACK 100H	Aloca memoria pentru stiva (stack) in modul real (real mode) din emu8086
.CODE	
MAIN PROC	Este o procedura definită de utilizator care poate fi folosită pentru a iniția execuția unui program.
PRINT "INPUT A CHARACTER: "	Afiseaza in consola mesajul INPUT A CHARACTER:

MOV AH, 1	Transferă valoarea 1 în registrul AH
INT 21H	Rulează funcția al cărui cod este stocat în registrul AH
MOV BL, AL	Asignează valoarea din registrul AL în registrul BL
CMP BL, "A"	Compara valoarea din registrul BL cu byteul 'A'
JNGE NONE	Este o instrucțiune de salt condițional, care se referă la instrucțiunea Jump if Not Greater or Equal. NONE este adresa de memorie la care sare dacă valoarea din registrul BL este mai mică ca 'A'.
CMP BL, "Z"	Compara valoarea din registrul BL cu byteul 'Z'
JNLE LOW	Este o instrucțiune de salt condițional, care se referă la instrucțiunea Jump if Not Lower or Equal. LOW este adresa de memorie la care sare dacă valoarea din registrul BL este mai mare ca 'Z'.
PRINTN	Afisează în consolă o linie nouă
PRINT "YOUR CASE : IS UPPER CASE!"	Afisează în consolă "YOUR CASE : IS UPPER CASE!"
JMP EXIT	Instrucțiunea JMP EXIT sare la eticheta EXIT.
LOW:	Adresa de memorie
CMP BL, "a"	Compara valoarea din registrul BL cu byteul 'a'
JNGE NONE	Este o instrucțiune de salt condițional, care se referă la instrucțiunea Jump if Not Greater or Equal. NONE este adresa de memorie la care sare dacă valoarea din registrul BL este mai mică ca 'a'.
CMP BL, "z"	Compara valoarea din registrul BL cu byteul 'z'
JNLE NONE	Este o instrucțiune de salt condițional, care se referă la instrucțiunea Jump if Not Greater or Equal. NONE este adresa de memorie la care sare dacă valoarea din registrul BL este mai mică ca 'z'.
PRINTN	Afisează în consolă o linie nouă
PRINT "YOUR CASE : IS LOWER CASE!"	Afisează în consolă "YOUR CASE : IS LOWER CASE!"
JMP EXIT	Instrucțiunea JMP EXIT sare la eticheta EXIT.

NONE:	Adresa de memorie
PRINTN	Afiseaza in consola o linie noua
PRINT "NONE OF THEM!"	Afiseaza in consola "NONE OF THEM!"
EXIT:	