

## Calcularea sumei unor numere dintr-un vector

1. Fie un vector cu N elemente. Calculati suma elementelor acestuia.

Ex: N=4

Valorile din vector: 1 , 5 , 2 , 7

```
segm_stiva    SEGMENT
```

```
    stiva     DW 128 DUP (?)
```

```
segm_stiva    ENDS
```

```
segm_date     SEGMENT
```

```
    rez       DB    ?
```

```
    vector    DB 4 DUP (1,5,2,7)
```

```
segm_date     ENDS
```

```
segm_cod      SEGMENT
```

```
    ASSUME CS:segm_cod, DS:segm_date, SS:segm_stiva
```

et:

```
        MOV AX, segm_date
```

```
        MOV DS, AX
```

```
        MOV SI, OFFSET vector
```

```
        MOV CX, SIZE vector + OFFSET vector
```

```
        MOV AL, 0
```

reia:

```

File Edit Format View Help
segm_stiva SEGMENT ; Directiva segment marcheaza inceputul segmentului

    stiva DW 128 DUP (?)

segm_stiva ENDS ; Directiva segment marcheaza sfarsitul segmentului

segm_date SEGMENT

    rez DB ? ; Declara un octet neinitializat
    vector DB 4 DUP (1,5,2,7) ; Declara si initializeaza elementele vectorului

segm_date ENDS

segm_cod SEGMENT

ASSUME CS:segm_cod, DS:segm_date, SS:segm_stiva ; Asociaza registrele cu segmentele

et:

    MOV AX, segm_date ; Se transfera adresele din segm_date in AX
    MOV DS, AX ; Se intoarneste continutul registrului DS cu continutul registrului AX

    MOV SI, OFFSET vector ; SI va fi contor de adrese, iar offset returneaza deplasarea variabilei fata de inceputul segmentului in care
    ; a fost definit
    MOV CX, SIZE vector + OFFSET vector ; In CX se incarca suma dintre dimensiunea vectorului si deplasarea variabilei
    MOV AL, 0 ; Muta in AL valoarea 0, initializandu-se in AL suma cu 0

reia:

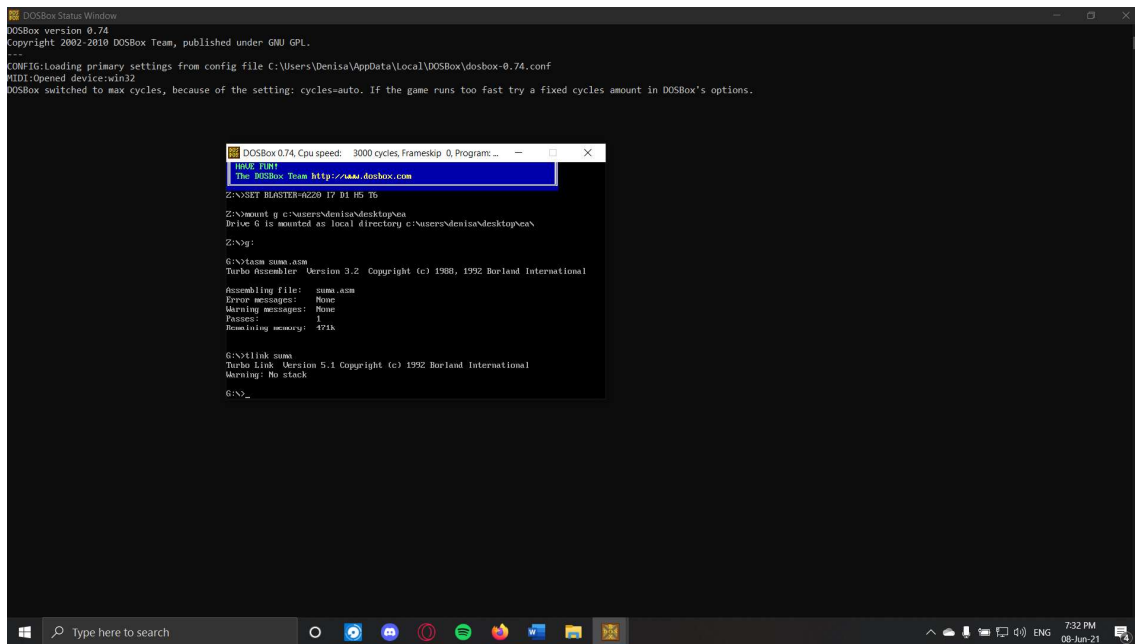
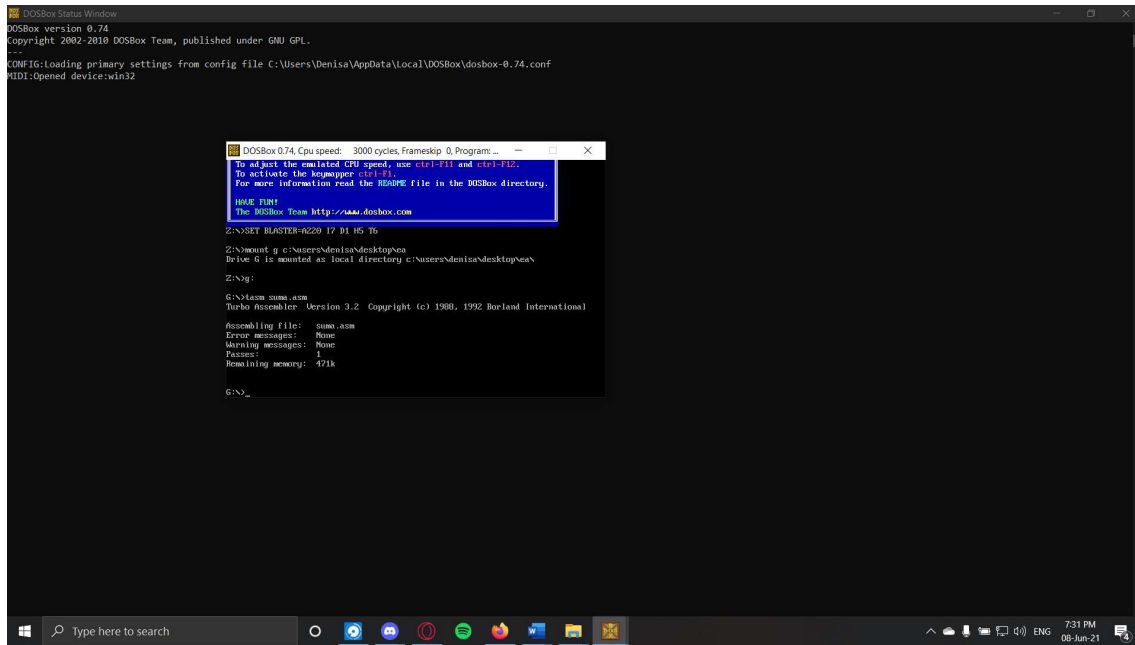
    ADD AL, BYTE PTR [SI] ; Aduna continutul octetului de memorie etichetat BYTE PTR la continutul lui AL, si va inlocui AL cu suma rezultata
    ADD SI, TYPE vector ; Pune in SI distanta asociata variabilei
    CMP SI, CX ; Compara valoarea din SI si CX
    JLE reia ; Daca nu sunt egale, executa JLE si sare la reia
    MOV rez, AL ; Valoarea lui AL este transferata in rez

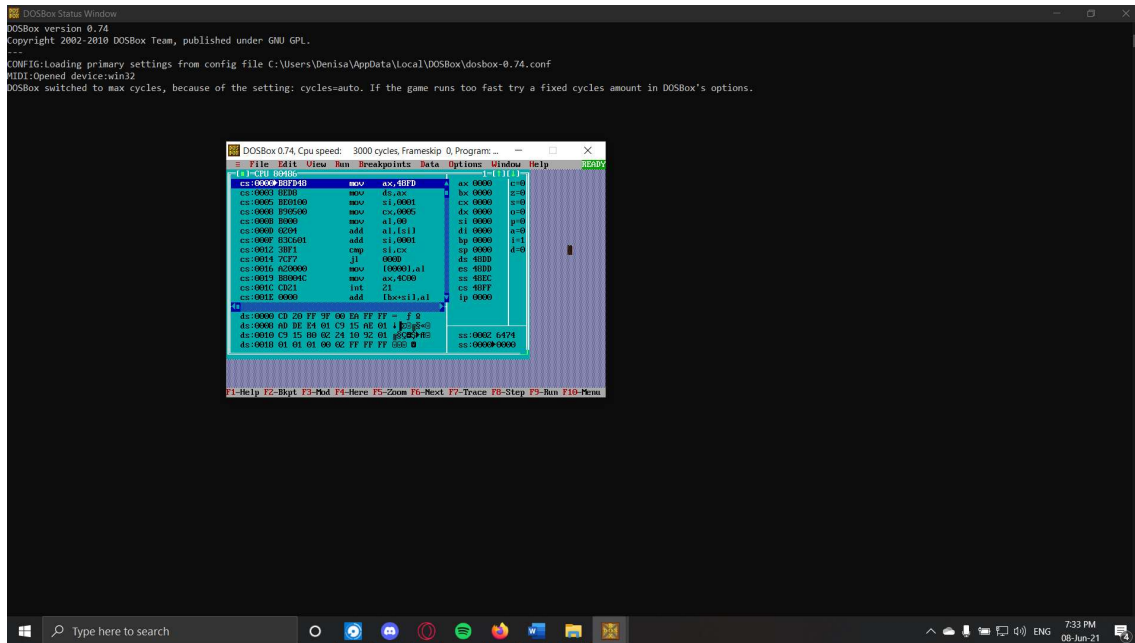
    MOV AX, 4C00H ; Registrul AX ia functia 4C00H, pentru terminarea executiei programului
    INT 21H ; Marcheaza finalul programului

segm_cod ENDS

END et

```





## Bibliografie:

- [https://elearning.unitbv.ro/pluginfile.php/3111378/mod\\_resource/content/0/Suport\\_Curs/S12Cap4Asamblare.pdf](https://elearning.unitbv.ro/pluginfile.php/3111378/mod_resource/content/0/Suport_Curs/S12Cap4Asamblare.pdf)
- <https://www.infoarena.ro/introducere-in-asamblare>
- <https://ocw.cs.pub.ro/courses/iocla/laboratoare/laborator-04>
- <http://webspace.ulbsibiu.ro/arpad.gellert/html/ASM.pdf>