

PROIECT BTI ASM

DINU RADU-GABRIEL

Grupa 1016. Seria C

1. Programul meu calculeaza minimul dintr-un vector dat, iar apoi executa dublarea lui.
2. Codul este urmatorul:

```
1 ;proiect - min elem uui vector, cu elem la nivel de W, fara
2 segm_stiva          SEGMENT
3     stiva           DW 128 DUP (?)
4 segm_stiva          ENDS
5
6 segm_date           SEGMENT
7     rez             DW ?
8     vector           DW 5 DUP (14,7,10,6,9)
9     msg              DB "proiect - gr1016, Dinu Radu-Gabriel!","$"
10
11 segm_date           ENDS
12
13 segm_cod             SEGMENT
14     ASSUME CS:segm_cod, DS:segm_date, SS:segm_stiva
15     et:
16     MOV AX, segm_date
17     MOV DS, AX
18
19     MOV SI, OFFSET vector
20     MOV CX, SIZE vector
21     MOV AX, [SI]
22
23     reia:
24     CMP [SI+2],AX
25     JG min
26     MOV AX, [SI+2]
27
28     min:
29     ADD SI, TYPE vector
30     CMP SI, CX
31     JL reia
32
33     ADD AX, AX
34     MOV rez, AX
35
36     MOV DX, OFFSET msg
37     MOV AH, 9
38     INT 21h
39
40     MOV AX, 4C00H
41     INT 21H
42 segm_cod             ENDS
43 END et
```

EXPLICATIA LINIILOR DE COD:

"segm_stiva SEGMENT" - Incepe un nou segment de memorie numit "segm_stiva"

"stiva DW 128 DUP (?)" - Declara un vector de 128 de elemente la nivel de W, fiecare element fiind initializat cu valoarea necunoscuta "?"

"segm_stiva ENDS" - Sfarseste segmentul de memorie "segm_stiva"

"segm_date SEGMENT" - Incepe un nou segment de memorie numit "segm_date"

"rez DW ?" - Declara o variabila numita "rez" care este initializata cu valoarea necunoscuta ?

"vector DW 5 DUP (14,7,10,6,9)" - Declara un vector de 5 elemente la nivel W, cu valorile 14, 7, 10, 6 si 9

"msg DB 'proiect - gr1016, Dinu Radu-Gabriel!','\$'" - Declara o variabila numita "msg" care contine textul "proiect - gr1016, Dinu Radu-Gabriel!" si caracterul de sfarsit de sir "\$"

"segm_date ENDS" - Sfarseste segmentul de memorie "segm_date"

"segm_cod SEGMENT" - Incepe un nou segment de memorie numit "segm_cod"

"ASSUME CS:segm_cod, DS:segm_date, SS:segm_stiva" - Stabileste ca segmentul de cod este "segm_cod", segmentul de date este "segm_date" si segmentul de stiva este "segm_stiva"

"MOV AX, segm_date" - Muta "segm_date" in registrul AX

"MOV DS, AX" - Muta valoarea din registrul AX in registrul DS

"MOV SI, OFFSET vector" - Muta pozitia inceputului vectorului "vector" in registrul SI

"MOV CX, SIZE vector" - Muta dimensiunea vectorului "vector" in registrul CX

"MOV AX, [SI]" - Muta valoarea din registrul SI in registrul AX

"CMP [SI+2], AX" - Compara valoarea de la pozitia SI + 2 registrul SI cu valoarea din registrul AX

"JG min" - Daca valoarea comparata este mai mare decat valoarea din registrul AX, salt la eticheta "min"

"MOV AX, [SI+2]" - Muta valoarea de la pozitia din registrul SI + 2 in registrul AX

"ADD SI, TYPE vector" - Adauga tipul elementului din vector la pozitia din registrul SI

"CMP SI, CX" - Compara valoarea din registrul SI cu valoarea din registrul CX

"JL reia" - Daca valoarea comparata este mai mica decat valoarea din registrul CX, salt la eticheta "reia"

"ADD AX, AX" - Adauga valoarea din registrul AX la sine insasi (o dubleaza)

"MOV rez, AX" - Muta valoarea din registrul AX in variabila "rez"

"MOV DX, OFFSET msg" - Muta pozitia inceputului (offset-ul) variabilei "msg" in registrul DX

"MOV AH, 9" - Seteaza registrul AH cu valoarea 9

"INT 21h" - Apelare interruptie sistem pentru afisarea textului din variabila "msg"

"MOV AX, 4C00H" - Seteaza registrul AX cu valoarea 4C00H (inchiderea programului)

"INT 21H" - Apelare interruptie sistem pentru a iesi din program

"segm_cod ENDS" - Sfarseste segmentul de memorie "segm_cod"

"END et" - Sfarseste programul et

3+4. Aici se poate observa rezultatul:

The screenshot displays an assembly project in Notepad++ with the following source code:

```
1 ;proiect - min elem uui vector, cu elem la nivel de W, fara LOOP, apoi executa dublarea acestuia
2 segm_stiva SEGMENT
3     stiva DW 128 DUP (?)
4 segm_stiva ENDS
5
6 segm_date SEGMENT
7     rez DW ?
8     vector DW 5 DUP (14,7,10,6,9)
9     msg DB "proiect - gr1016, Dinu Radu-Gabriel!", '$'
10
11 segm_date ENDS
12
13 segm_cod SEGMENT
14     ASSUME CS:segm_cod, DS:segm_date, SS:segm_stiva
15     et:
16     MOV AX, segm_date
17     MOV DS, AX
18
19     MOV SI, OFFSET vector
20     MOV CX, SIZE vector
21     MOV AX, [SI]
22
23     reia:
24     CMP [SI+2], AX
25     JG min
26     MOV AX, [SI+2]
27
28     min:
29     ADD SI, TYPE vector
30     CMP SI, CX
31     JL reia
32
33     ADD AX, AX
34     MOV rez, AX
35
```

Overlaid on the code is a DOSBox emulator window showing the execution of assembly instructions. The instruction list includes:

- cs:0000:00000 mov si,0000
- cs:0000:00000 mov cx,0000
- cs:0000:00000 mov ax,1411
- cs:0000:294402 cmp [si+02],ax
- cs:0010:7703 ja 0015
- cs:0012:884402 mov ax,[si+02]
- cs:0015:633602 add si,0002
- cs:0018:38f1 cmp si,cx
- cs:001a:7c71 jl 0000
- cs:001c:6030 add ax,ax
- cs:001e:a30000 mov 100001,ax
- cs:0021:b03400 mov dx,0034
- cs:0024:8409 mov al,09
- cs:0026:c021 int 21
- cs:0028:b0004c mov ax,4000

Below the DOSBox window is a Notepad window titled "rezProiect.txt" containing the output of the program:

```
min(14,7,10,6,9) = 6 = (06)H
6*2 = 12 = (0C)H
AX: 000C
mem: 0C 00
```