1. (30 minute) Se dă listingul rezultat după rularea asamblorului pe următorul program care face un calcul asupra a 3 numere:

|  |  |  |
| --- | --- | --- |
|  | | .586  .MODEL FLAT  .STACK 4096 ; rezervă o stivă de 4096 de bytes |
|  | |  |
| 00000000 | | .DATA ; rezervă o zonă de memorie pentru date |
| 00000000 | 00000055 | X DD 85 |
| 00000004 | FFFFFFD1 | Y DD -47 |
|  | |  |
| 00000008 | \_\_\_\_**(a)**\_\_\_\_ | Z DD 91 |
|  | |  |
| \_\_\_\_**(b)**\_\_\_\_ | 00000000 | Calcul DD 0 |
|  | |  |
| 00000000 | | .CODE ; rezervă o zonă de memorie pentru program |
|  | |  |
| 00000000 | | main proc |
|  | |  |
| 00000000 | A1 00000000 R | MOV EAX, [X] ; pune primul număr în EAX |
| \_\_\_\_**(c)**\_\_\_\_ | 2B 05 00000004 R | SUB EAX, [Y] ; scade al doilea număr din el |
| 0000000B | 1B 05 00000008 R | SBB EAX, [Z] ; scade al treilea număr din el |
| 00000011 | A3 05 0000000C R | MOV [Calcul], AX ; pune rezultatul în memorie |
|  | |  |
| 00000016 | B8 00000000 | MOV EAX, 0 ; ieşire cu codul de retur 0 |
| 0000001B | C3 | RET |
|  | |  |
| 0000001C | | Main endp |
|  | |  |
|  | | END ; end of source code |

1. Care este dimensiunea programului în octeţi? Răspundeţi în zecimal.
2. Completaţi cu valorile corecte la punctele a), b) şi c)
3. Care este OPCODE-ul instrucţiunii ADD EAX, [Z]?
4. Care este codul obiect al instrucţiunii ADD EAX, [Z]?
5. Care este codul obiect al instrucţiunii MOV EAX, [X]?
6. Identificaţi 3 directive de asamblor în acest program

Observaţie: notaţia  R provine de la "relocatable" şi înseamnă că asamblorul construieşte blocuri de cod de atomicitate variabilă care pot fi plasate în orice regiune a memoriei unde linker-ul găseşte loc disponibil.