**ACSL Assembly Language**

Este un limbaj de programare simplificat - este un procesor teoretic care are un singur registru.

Exemple

1) Fie următorul progrămel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A DC 5  LOAD A  ADD =1  STORE A | Iată efectul instrucțiunilor:   |  |  | | --- | --- | | A | ACC | | 5  6 | 0  5  6 | |

Concluzie: programul de mai sus face de fapt **A++**

(incrementează valoarea variabilei A cu 1)

2)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| READ A  LOAD A  DIV =2  MULT =2  SUB A  BE CAZUL1 (\*)  PRINT "IMPAR"  BU GATA  CAZUL1 PRINT "PAR"  GATA END  Obs: (\*) Dacă în urma scăderii SUB A rezultatul a devenit 0, atunci se sare necondiționat la CAZUL1 - și dei se afișează "PAR" | Să analizăm efectul instrucțiunilor:  Să luăm niște exemple  A=3   |  |  | | --- | --- | | A | ACC | | 3 | 3  1  2  -1 |   A=4   |  |  | | --- | --- | | A | ACC | | 4 | 4  2  4  0 | |

Iată ce program complicat DOAR pentru a face ceea ce în C++ se face prin:

**if(a%2==0) cout<<"PAR";**

**else cout<<"IMPAR";**

3)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| READ N I DC 1  ETICHETA1 PRINT I  LOAD I  SUB N  BE GATA  LOAD I  ADD =1  STORE I  BU ETICHETA1  GATA END | Să analizăm efectul instrucțiunilor:  Fie N=3   |  |  |  |  | | --- | --- | --- | --- | | N | I | ACC | Ce se afișează | | 3 | 1  2  3 | 1  -2  1  2  -1  2  3  0 | 1  2  3 | |

Acest program este de fapt echivalent cu

**for(i=1;i<=n;i++)cout<<i;**