

## Problema 1.

L'objectiu d'aquest problema consisteix en traduir fragments de codi en C a llenguatge ensamblador.

En els codis escrits en C no declarem les variables que es fan servir. Suposem que aquestes variables ja les tenim declarades i en tots els casos són nombres **enters sense signe de 32 bits** codificats en complement a dos. Tampoc heu de reservar espai en memòria ni inicialitzar les variables en la memòria de l'ensamblador. Per referir-nos a l'adreça de memòria d'una variable que surt en el codi en C utilitzem el mateix nom de la variable en C. Per exemple, per traduir la següent sentència en C,

**A = A + B;**

podem fer servir el codi següent:

```
LLA R1, A
LW R2, 0(R1)
LLA R3, B
LW R3, 0(R3)
ADD R3, R2, R3
SW R3, 0(R1)
```

Traduiu a llenguatge ensamblador cadascun dels següents fragments de codi.

**a) A=B+C\*D;**

**&B significa l'adreça de la variable B**

LLA R1, B ; R1 ← &B, R1 emmagatzema la direcció de la variable B  
LW R1, 0(R1) ; R1 ← [mem (0+R1)], per tant R1 emmagatzema la dada de  
; la variable B. R1 ← [B]

**[B] significa el contingut de la variable B (la seva adreça està en R1)**

LLA R2,C ; R1 ← &C,  
LW R2, 0(R2) ; R2 ← [C], R2 emmagatzema la dada de la variable C  
LLA R3,D ; R3 ← &D,  
LW R3, 0(R3) ; R3 ← [D], R3 emmagatzema la dada de la variable D  
MUL R2,R2,R3 ; R2 ← R2\*R3 , R2 ← C\*D  
ADD R1,R1,R2 ; R1 ← R1+R2 , R1 ← B + C\*D  
LLA R2,A ; R2 ← &A,  
SW R1,0(R2) ; [A] ←R1, Guardem en A el resultat de B+C\*D