

**Contrôle continu**  
**Programmation des systèmes**

**Les réponses aux questions doivent être commentées clairement et lisiblement.**

**Exercice 1:** Ecrire un programme C pour la Gameboy advance qui affiche toutes les 64 secondes la lettre A à l'écran. L'utilisateur peut changer de lettre et afficher la lettre A ou B en pressant la touche A ou B du clavier. Pour l'affichage vous n'avez pas besoin de coder la fonction, le clavier peut-être accédé par scrutation (pas d'interruption).

**Exercice 2:** Un algorithme efficace<sup>1</sup> pour multiplier deux nombres x et y est:

```
r=0
while(x !=0)
    if (x mod 2 == 1)
        r=r+y
    x=x/2
    y=y*2
```

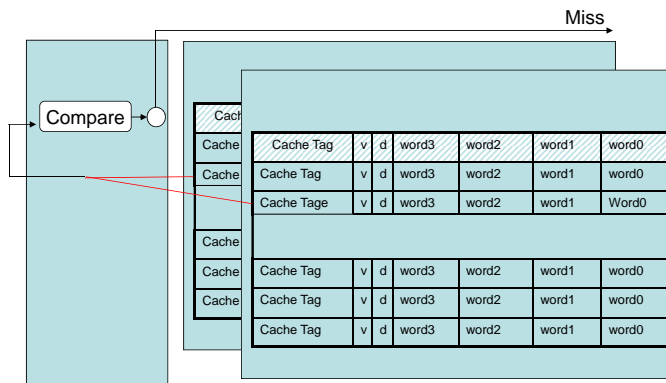
Ecrivez le programme assembleur. Supposez que x est dans r0, y dans r1 et le résultat r dans r2. On suppose que x>0 au début de l'exécution (utiliser une boucle repeat-until est correct aussi).

**Exercice 3:** On considère une structure de mémoire cache composée de deux structures de mémoires caches à accès directs dont les Cache Tags sont stockés dans une mémoire associative, voir la figure ci-dessous.

- Si la capacité totale de la mémoire cache est de 4Kb quelle est la décomposition de l'adresse pour accéder à la mémoire cache?
- Décrivez l'utilité des bits v et d.
- Donnez les algorithmes de lecture et écriture si la mémoire cache est *read-allocate* et *writethrough*.

---

<sup>1</sup>Bien sûr ici utiliser l'instruction MUL est le plus efficace. Mais pour implémenter MUL cet algorithme est plus efficace que celui que vous avez appris à l'école.



**Exercice 4:** La mémoire ROM d'une GBA contient les instructions suivantes.

```

00000018  b      0x128

00000128  stmfd  r13!, {r0-r3,r12,r14}
0000012C  mov    r0, 0x4000000
00000130  add    r14, r15,0x0
00000134  ldr    r15, [r0, -4]
00000138  ldmdf  r13!, {r0-r3,r12,r14}
0000013C  subs   r15, r14, #4

```

Décrivez précisément la fonction implémentée par chaque instruction (le contenu des registres accédés et modifiés) ainsi que la fonction du code.