

5 – Révisions

1.1 Exercice de programmation

Traduire en assembleur le programme suivant :

```
int X[10]={2, 4, 6, 8, 10, 12, 14,16, 18, 20}  
for (i=0; i<10; i++) X[i] = X[i]/2 ;
```

1.2 Exercice de programmation

Soient les nombres de trois chiffres abc dont on cherche à tester la propriété suivante:

$$a^3 + b^3 + c^3 = abc.$$

Par exemple : $3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$. L'objectif est d'écrire un programme qui réalise ce test, et qui sera structuré en 3 sous-programmes : `TEST_CUBE`, `VALEUR_NOMBRE` et `SOMME_CUBE`.

Ecrire, en respectant les contraintes qui vous sont imposées :

- Un programme principal appelant une fonction `TEST_CUBE` qui reçoit ses paramètres par la pile. Le programme principal utilisera trois variables `A`, `B` et `C` pour représenter respectivement les centaines, les dizaines et les unités du nombre (que l'on supposera déjà décomposé) et seront initialisées de façon à traiter l'exemple 371. Le résultat du test sera renvoyé dans une variable `TEST_RES` (0 si le test est faux, 1 si le test est vérifié). Toutes les variables manipulées seront de type `.word`.
- Un sous-programme `TEST_CUBE` qui reçoit ses paramètres par la pile et renvoie dans le registre `R0` le résultat du test (0 si le test est faux, 1 si le test est vérifié). Ce sous-programme devra faire appel à deux sous-programmes `VALEUR_NOMBRE` et `SOMME_CUBE`.
- Un sous-programme `VALEUR_NOMBRE` qui reçoit ses paramètres par les registres `R1`, `R2`, `R3` (respectivement centaines, dizaines, unités) et renvoie dans le registre `R0` la valeur décimale du nombre abc . Cette fonction ne devra utiliser aucune instruction de multiplication.
- Un sous-programme `SOMME_CUBE` qui reçoit ses paramètres par les registres `R1`, `R2`, `R3` (respectivement centaines, dizaines, unités) et renvoie dans le registre `R0` le résultat de $a^3 + b^3 + c^3$.