

Programmation en assembleur ARM

Calcul sur des registres

- ★

Ecrire un programme qui calcule la somme des 49 premiers entiers.

 1
- ★

Ecrire un programme qui calcule (de manière itérative) la factorielle d'un nombre rangé dans r0.

 2
- ★

On suppose que r0 contient un caractère représentant une opération ('+', '-' ou '*') et que r1 et r2 contiennent deux valeurs entières. Ecrire un programme qui range dans r3 le résultat de l'opération indiquée par r0 appliquée aux opérandes r1 et r2.
Par exemple, si r0 contient '*', r1 contient 3 et r2 contient 4, le programme doit écrire 12 dans r3.

 3
- ★

Ecrire un programme qui calcule les sommes des N premiers entiers :

 - multiples de 2 (dans r0)
 - multiples de 4 (dans r1)
 - multiples de 8 (dans r2).

 4
- ★★

Ecrire un programme qui calcule la somme N premières puissances de 2 à partir de 2¹.

 5
- ★★

Ecrire un programme qui calcule le quotient (dans r2) et le reste (dans r3) de la division entière de r0 par r1.

 6
- ★★★

Utiliser des instructions conditionnelles (pas de branchement) pour écrire un programme qui réalise :
r2 = MAX(r0,r1)

 7
- ★★★

Ecrire une séquence d'instructions qui multiplie par 10 le contenu de r0 sans utiliser d'instruction de multiplication.

 8
- ★★★★

Ecrire un programme qui détermine si la *représentation binaire* d'un nombre entier rangé dans r0 est un palindrome.

 9
- ★★★★

Ecrire une séquence d'instructions qui multiplie par 35 le contenu de r0 sans utiliser d'instruction de multiplication.

 10

Accès à la mémoire, calcul sur des tableaux

- ★

Ecrire un programme qui initialise les éléments d'un tableau de 10 octets avec les 10 premiers entiers.
Proposer deux versions :

 - une qui référence un élément à partir d'un pointeur qui balaye le tableau
 - l'autre qui référence un élément à partir de son numéro et de l'adresse de base du tableau

 11
- ★★

Ecrire un programme qui recherche le maximum dans un tableau de 10 entiers signés. Utiliser le mode d'adressage indexé (adresse calculée à partir de l'adresse de base et numéro d'élément). Le résultat est le numéro de l'élément le plus grand (le premier si plusieurs éléments ont la valeur du maximum).

 12
- ★★

Ecrire un programme qui initialise les éléments d'un tableau, à partir de tab[2], de la manière suivante :
tab[i] = tab[i-1] + tab[i-2]

 13
- ★★

Ecrire un programme qui compte (dans r0) le nombre de 'a' dans une chaîne de caractères et place dans r1 la position du premier 'a'.

 14

★★★	<p>Ecrire un programme qui recopie le contenu d'une zone mémoire dans une autre. Les adresses respectives des zones sont dans r1 (zone de départ) et r2 (zone d'arrivée). La taille d'une zone est de N octets. Envisager tous les cas possibles : les zones se chevauchent ou pas, la zone de départ est située plus bas ou plus haut que la zone d'arrivée dans la mémoire.</p> <p>Dans un premier temps, faire des transferts octet par octet, puis optimiser le code en transférant des mots, des demi-mots et des octets de manière à minimiser le nombre de transferts.</p>	15
★★★	<p>On considère une matrice de dimension $m \times n$ rangée en mémoire à l'adresse MAT. En supposant que r0 et r1 contiennent les numéros de ligne et de colonne i et j, écrire le code (limité à 5 instructions) permettant d'initialiser à 0 l'élément MAT[i,j] :</p> <ul style="list-style-type: none"> • si les éléments sont des octets • si les éléments sont des mots longs (en ne modifiant qu'une instruction) 	16
★★★	<p>Ecrire un programme qui transforme une chaîne de caractères en une valeur entière. On suppose que tous les caractères sont des chiffres, et que la chaîne se termine par le code ASCII 0.</p>	17
★★★★	<p>Ecrire un programme qui transforme une valeur entière positive, codée sur 32 bits, en une chaîne de N+1 caractères (permettant de représenter une valeur décimale sur N chiffres, le dernier caractère devant être un 0).</p>	18

Sous-programmes

★	<p>Ecrire un sous-programme qui initialise à 0 tous les éléments (entiers) d'un vecteur dont l'adresse TAB et la taille N sont connues statiquement. Faire en sorte que ce sous-programme restaure, quand il se termine, les valeurs contenues dans les registres lors de son appel.</p> <p>Modifier le sous-programme de telle sorte que la valeur d'initialisation des éléments du tableau lui soit passée en paramètre par le registre r0.</p> <p>Ecrire un programme principal qui appelle ce sous-programme.</p>	19
★	<p>Ecrire un sous-programme auquel on passe en paramètre, par registre, l'adresse d'une chaîne de caractères et un caractère, et renvoie dans r0 un booléen qui indique si le caractère est présent dans la chaîne ou non.</p> <p>Ecrire un programme principal qui appelle ce sous-programme.</p>	20
★★	<p>Ecrire un sous-programme qui reçoit deux paramètres et renvoie dans r0 le maximum des deux paramètres :</p> <ul style="list-style-type: none"> • avec des paramètres passés par registres • avec des paramètres passés par la pile <p>Ecrire un programme principal qui appelle ce sous-programme.</p>	21
★★	<p>Ecrire un sous-programme qui initialise à 0 une zone mémoire dont l'adresse et la taille en octets sont passés par la pile.</p> <p>Ecrire un programme principal qui appelle ce sous-programme pour initialiser un tableau de N octets.</p>	22
★★★	<p>Ecrire :</p> <ul style="list-style-type: none"> • un sous-programme AJOUTE_UN qui reçoit en paramètre par la pile un nombre X et renvoie dans le registre r0 la valeur X+1. • un sous-programme COMPLEMENT_A_2 qui reçoit en paramètre, dans le registre r0, un nombre Y et renvoie, dans le registre r1, le complément à 2 de Y. Ce complément à 2 sera calculé en inversant Y (l'instruction mvn r3,r4 permet de mettre dans le registre r3 le complément à 1 de r4) puis en appelant le sous-programme AJOUTE_UN. • le programme principal qui appelle le sous-programme COMPLEMENT_A_2 pour calculer le complément à 2 d'une valeur présente dans le registre r0. Au retour du sous-programme, on veut que r0 contienne toujours la valeur de départ. Le résultat sera récupéré dans r1. <p>Faire la trace de l'exécution de ce programme.</p>	23
★★★★	<p>Ecrire un sous-programme récursif qui calcule la factorielle d'un nombre passé en paramètre par la pile. Le résultat sera renvoyé dans r0. Ecrire un programme principal qui appelle ce sous-programme.</p> <p>Faire la trace de l'exécution de ce programme lorsqu'il calcule 4! .</p>	24