

UNIVERSITE Joseph FOURIER
Polytech' Grenoble

ALM (Soft) : Travail pratique No 2 Interclassement de tableau en assembleur ARM

Compléter le programme assembleur ARM suivant qui effectue l'interclassement de deux tableaux T1 et T2 et affiche le résultat a l'écran.

Rappel : pour afficher un entier relatif de 32 bits on pourra utiliser la fonction EcrRelatif32 (contenu dans le fichier es.s) selon le mode d'emploi suivant :

```
mov R1, ... @ on met dans R1 la valeur a afficher
bl EcrRelatif32 @ appel a la fonction d'écriture
```

Référez-vous au premier TP pour compiler, exécuter et éventuellement déboguer votre programme.

```
@ -----
@ Algorithmme
@ -----

@ lexique :

@ N1 : la constante entiere 5
@ N2 : la constante entiere 4
@ N : la constante entiere N1+N2
@ T1 : un tableau sur 0..N1-1 d'entiers relatifs sur 8 bits, trie decroissant
@ T1 = [12, 10, 8, 8, -7]
@ T2 : un tableau sur 0..N2-1 d'entiers relatifs sur 16 bits, trie decroissant
@ T2 = [15, 8, -1, -4]
@ T : un tableau sur 0..N1+N2-1 d'entiers relatifs sur 32 bits
@
@ interclasser : l'action
@ -- calcule dans dans T l'interclassement des tableaux T1 et T2
@ i1, i2, i : des entiers naturels -- indice de parcours de T1, T2, et T
@ debut
@
@ i1 <- 0 ; i2 <- 0 ; i <- 0
@ tantque (i1<N1)
@ tantque (i2<N2) etpuis (T2[i2]>=T1[i1])
@ T[i] = T2[i2]
@ i <- i+1          -- on avance sur T
@ i2 <- i2+1       -- on avance sur T2
@ ftantque
```

```

@ T[i] = T1[i1]
@ i <- i+1      -- on avance sur T
@ i1 <- i1+1   -- on avance sur T1
@ ftantque
@
@ -- i1=N1, il faut maintenant recopier dans T la fin du tableau T2
@ tantque (i2<N2)
@ T[i] = T2[i2]
@ i <- i+1      -- on avance sur T
@ i2 <- i2+1   -- on avance sur T2
@ ftantque
@
@ -- affichage du resultat
@ i <- 0
@ tantque (i<N)
@ afficher(T[i])
@ i <- i+1      -- on avance sur T
@ ftantque
@ fin

@ -----
@ Assembleur ARM
@ -----

.data
.set N1, 5 @ definition des constantes N, N1 et N2
.set N2, 4
.set N, N1+N2

T1:
@ declaration et initialisation de T1, a completer ...

T2:
@ declaration et initialisation de T2, a completer ...

T:
@ declaration de T, a completer ...

.text

.global main

main: STMFd sp!, {lr} @ ne pas modifier

@ inserer ici le code de "interclasser"

fin: LDMFD sp!, {pc} @ ne pas modifier

relaisT1: .word T1 @ definition des variables relais pour T1, T2 et T
relaisT2: .word T2
relaisT: .word T

```