```
@ Convention d'appel avec les paramètres passés dans des registres

@ Max2
@    Convention d'appel   x dans r0, y dans r1, maxi dans r2 ( et adresse de ret
our dans lr)
@    registres modifiés : variable locale rmax : r8

@ calcul
@    Convention d'appel   (adresse de retour dans lr)
@    registres modifiés : variableslocales e,f,max_ef : r4,r5,r6
@                         paramètres passés à max2 :    r0,r1,r2

@ Une zone mémoire locale par fonction (non récusive) affectée statiquement dans
 bss

               .bss
                          @ pour sauvegarde des registres modifiés dans calcul
CALCUL_DELTA_R0=-24
CALCUL_DELTA_R1=-20
CALCUL_DELTA_R2=-16
CALCUL_DELTA_R4=-12
CALCUL_DELTA_R5=-8
CALCUL_DELTA_R6=-4
CALCUL_DELTA_LR=-0

               .skip 4     @ sauvegarde r0  -24
               .skip 4     @ sauvegarde r1  -20
               .skip 4     @ sauvegarde r2  -16
               .skip 4     @ sauvegarde r4  -12
               .skip 4     @ sauvegarde r5   -8
               .skip 4     @ sauvegarde r6   -4
local_calcul:  .skip 4     @ sauvegarde lr   -0


                          @ pour sauvegarde des registres dans max2
MAX2_DELTA_R8=-0

local_max2:    .skip 4     @ r8  -0

               .global     calcul
               .text

@ Convention d'appel de max2
@ x : r0  y :r y  maxi : r2   adresse retour : lr
@
@ Variable locale rmax : r8

max2:                        @ prologue : sauvegarde de r8
         ldr  fp,=local_max2
         str  r8,[fp,#MAX2_DELTA_R8]

                             @ corps inchangé
         mov  r8,r0          @ rmax=x

if1:     cmp  r1,r8          @ if (y ≤ rmax) goto finsi1
         bls  finsi1
alors1:  mov  r8,r1          @ rmax = y
finsi1:
         str  r8,[r2]        @ *maxi = rmax

                             @ epilogue : restauration et retour
         ldr  fp,=local_max2
```



```
         ldr  r8,[fp,#MAX2_DELTA_R8]
         mov  pc,lr

@ Convention d'appel de calcul
@ adresse de retour : lr
@
@ Variables locales e : r4  f: r5  max_ef : r6

#define e r4
#define f r5
#define max_ef r6

calcul :                     @ prologue  : sauvegarder  les registres
         ldr   fp,= local_calcul
         str   lr,[fp,#CALCUL_DELTA_LR]
         str   r6,[fp,#CALCUL_DELTA_R6]
         str   r5,[fp,#CALCUL_DELTA_R5]
         str   r4,[fp,#CALCUL_DELTA_R4]
         str   r2,[fp,#CALCUL_DELTA_R2]
         str   r1,[fp,#CALCUL_DELTA_R1]
         str   r0,[fp,#CALCUL_DELTA_R0]

         mov  r4,#5          @ e=5
         mov  r5,#6          @ f=6

         mov  r6,r4      @ max_ef=e

if2:     cmp  r5,r6      @ if (f ≤ max_ef) gotofinsi2
         bls  finsi2
alors2:  mov  r6,r5      @ max_ef = f
finsi2:
                             @ max2(a,b,&max_ab)
         ldr  r0,=a          @ x = a
         ldr  r0,[r0]

         ldr  r1,=b          @ y = b
         ldr  r1,[r1]

         ldr  r2,=max_ab     @ maxi=&max_ab

         bl   max2

                             @ max2(a,b,&max_ab)
         ldr  r0,=c          @ x = c
         ldr  r0,[r0]

         ldr  r1,=d          @ y = d
         ldr  r1,[r1]

         ldr  r2,=max_cd     @ maxi=&max_cd

         bl   max2

                             @ epilogue : restaurer lr + retour
         ldr   fp,= local_calcul
         ldr   r0,[fp,#CALCUL_DELTA_R0]
         ldr   r1,[fp,#CALCUL_DELTA_R1]
         ldr   r2,[fp,#CALCUL_DELTA_R2]
         ldr   r4,[fp,#CALCUL_DELTA_R4]
         ldr   r5,[fp,#CALCUL_DELTA_R5]
         ldr   r6,[fp,#CALCUL_DELTA_R6]
         ldr   lr,[fp,#CALCUL_DELTA_LR]
```

```
        mov  pc,lr

        .ltorg
```