

L'ASSEMBLEUR – for et while

Xavier Merrheim

# Traduire un for

```
int i,s;  
main()  
{  
s=0;  
for(i=1;i<10;i++)s=s+i;  
}
```

L1:

```
.word i  
.word s  
.comm i,4,4  
.comm s,4,4  
mov r0,#0  
ldr r1,L1+4  
str r0,[r1]  
mov r0,#1  
ldr r1,L1  
str r0,[r1]  
for: ldr r0,L1  
ldr r1,[r0]  
cmp r1,#10
```

## Traduction

```
bge fin  
ldr r0,L1+4  
ldr r1,[r0]  
ldr r2,L1  
ldr r3,[r2]  
add r1,r1,r3  
str r1,[r0]  
ldr r0,L1  
ldr r1,[r0]  
add r1,r1,#1  
str r1,[r0]  
bra for  
fin:
```

# Exercice : traduire en assembleur ARM

```
int a,b,s,i;  
main()  
{a=1;b=2;s=0;  
for(i=1;i<=12;i++)  
    {  
    a=a+b;  
    b=b+1;  
    s=s+a;  
    }  
}
```

# Solution

L1:	mov r0,#0	str r1,[r0]	str r1,[r0]
.word a	ldr r1,L1+8	ldr r0,L1+4	bra for
.word b	str r0,[r1]	ldr r1,[r0]	fin:
.word s	mov r0,#1	add r1,r1,#1	
.word i	ldr r1,L1+12	str r1,[r0]	
.comm a,4,4	str r0,[r1]	ldr r0,L1+8	
.comm b,4,4	for: ldr r0,L1+12	ldr r1,[r0]	
.comm s,4,4	ldr r1,[r0]	ldr r2,L1	
.comm i,4,4	cmp r1,#12	ldr r3,[r2]	
mov r0,#1	bgt fin	add r1,r1,r3	
ldr r1,L1	ldr r0,L1	str r1,[r0]	
str r0,[r1]	ldr r1,[r0]	ldr r0,L1+12	
mov r0,#2	ldr r2,L1+4	ldr r1,[r0]	
ldr r1,L1+4	ldr r3,[r2]	add r1,r1,#1	
str r0,[r1]	add r1,r1,r3		

# Exercice : Traduire en assembleur ARM

```
int a,b;  
main()  
{  
a=1;b=2;  
while(a<100)  
    {  
    b=a+b;  
    a=a+b;  
    }  
    a=a+10;  
}
```

# Solution

```
L1:
.word a
.word b
.comm a,4,4
.comm b,4,4
mov r0,#1
ldr r1,L1
str r0,[r1]
mov r0,#2
ldr r1,L1+4
str r0,[r1]
```

```
while: ldr r0,L1
      ldr r1,[r0]
      cmp r1,#100
      bge fin
      ldr r0,L1
      ldr r1,[r0]
      ldr r0,L1+4
      ldr r2,[r0]
      add r1,r1,r2
      str r1,[r0]
      ldr r0,L1
      ldr r1,[r0]
```

```
      ldr r2,L1+4
      ldr r3,[r2]
      add r1,r1,r3
      str r1,[r0]
      bra while
fin : ldr r0,L1
      ldr r1,[r0]
      add r1,r1,#10
      str r1,[r0]
```

# EXERCICE

```
Int a,b,c,d;  
main()  
{a=10;b=1;c=2;d=3;  
while(a+b>c+d)  
    {c=c+1;  
    d=d+c;  
    }  
a=a+c-d;  
}
```



L1.			
.word a	mov r0,#2	add r2,r2,r3	Fin: ldr r0,L1
.word b	ldr r1,L1+8	cmp r1,r2	ldr r1,[r0]
.word c	str r0,[r1]	ble fin	ldr r2,L1+8
.word d	mov r0,#3	ldr r0,L1+8	ldr r3,[r2]
.comm a,4,4	ldr r1,L1+12	ldr r1,[r0]	add r1,r1,r3
.comm b,4,4	str r0,[r1]	add r1,r1,#1	ldr r2,L1+12
.comm c,4,4	while:ldr r0,L1	str r1,[r0]	ldr r3,[r2]
.comm d,4,4	ldr r1, [r0]	ldr r0,L1+12	sub r1,r1,r3
mov r0,#10	ldr r0,L1+4	ldr r1,[r0]	str r1,[r0]
ldr r1,L1	ldr r2,[r0]	ldr r2,L1+8	
str r0,[r1]	add r1,r1,r2	ldr r3,[r2]	
mov r0,#1	ldr r0,L+8	add r1,r1,r3	
ldr r1,L1+4	ldr r2, [r0]	str r1,[r0]	
str r0,[r1]	ldr r0,L1+12	bra while	
	ldr r3,[r0]		