

nov. 05, 18 14:13	document_td_procedure_pile.latin1.txt	Page 1/7
<pre> int g_moins_d; /* retourne f* pgcd */ /* calcule difference et de max a et b */ int npgcd_diff (int f, int a, int b, int *a_moins_b, int *max) { int d; d = a - b; *a_moins_b = d; *max = a; if (b>a) *max= b; while (a != b) if (a > b) a = a - b; else b = b - a; return (f*a); } /* Calcul de somme, difference, min, max, n x pgcd, de x et y */ void calcul (int x, int y, int *plus, int *moins, int *np, int n, int *troismax) { register int calc; int m; calc = npgcd_diff (n,y,x,moins,&m); *np = calc; calc = x + y; *plus = calc; m = m + (m << 1); *troismax = m; } int main () { int somme, diff, pg,t; register int gauche = 5; register int droit = 15; calcul (gauche, droit, &somme, &diff,&pg,10,&t); printf ("x = %d y = %d somme= %d diff = %d pgcd X 10 = %d 3max = %d\n", gauche,droit,somme,diff,pg,t); } .global main main: mov ip, sp stmfd sp!, {fp, ip, lr, pc} sub fp, ip, #4 @ Utilisation des registres r4 a r10 @ gauche : r4 @ droit : r5 @ temporaire contenu : r6 @ temporaire adresse : r7 </pre>		

nov. 05, 18 14:13	document_td_procedure_pile.latin1.txt	Page 2/7
<pre> @ espace local @ registres de travail @ sauvegarde registre r4 : fp -16 @ sauvegarde registre r5 : fp -20 @ sauvegarde registre r6 : fp -24 @ sauvegarde registre r7 : fp -28 @ variable locale somme : fp -32 @ variable locale diff : fp -36 @ variable locale pg : fp -40 @ variable locale t : fp -44 sub sp, sp, #44 @ 8 locaux et 3 parametres passe s str r4, [fp, #-16] str r5, [fp, #-20] str r6, [fp, #-24] str r7, [fp, #-28] mov r4, #5 @ gauche = 5 mov r5, #15 @ droit = 15 @ parametres d'appel de calcul mov r0, r4 @ x = gauche dans r0 mov r1, r5 @ y = droit dans r1 sub r2, fp, #32 @ plus = &somme dans r2 sub r3, fp, #36 @ moins = &diff dans r3 sub r7, fp, #44 @ r7 = &t str r7, [sp, #8] @ troismax = &t dans pile ==> sp + 8 mov r6, #10 @ n = 10 dans pile ==> sp + 4 str r6, [sp, #4] @ r7 = &pg sub r7, fp, #40 @ np = &pg dans pile ==> sp str r7, [sp] bl calcul @ parametres d'appel de printf ldr r0, =format @ param1 : format dans r0 mov r1, r4 @ param2 : gauche dans r1 mov r2, r5 @ param3 : droit dans r2 ldr r3, [fp, #-32] @ param4 : valeur de somme dans r3 ldr r6, [fp, #-44] @ r7 = *&t str r6, [sp, #8] @ param7 : t dans la pile ==> sp + 8 ldr r6, [fp, #-40] @ r6 = *&pg str r6, [sp, #4] @ param6 : pg dans la pile ==> sp +4 ldr r6, [fp, #-36] @ r6 = *&diff str r6, [sp] @ param5 : diff dans la pile ==> sp bl printf ldr r7, [fp, #-28] ldr r6, [fp, #-24] ldr r5, [fp, #-20] ldr r4, [fp, #-16] ldmea fp, {fp,sp,pc} .data format: .asciz "x= %d y = %d somme = %d diff = %d pgcdx10 = %d 3max=%d\ 012" </pre>		

nov. 05, 18 14:13	document_td_procedure_pile.latin1.txt	Page 3/7
npgcd_diff:	<pre> .text .global npgcd @ f : r0 @ a : r1 @ b : r2 @ a_moins_b : r3 @ sp pointe sur max @ resultat retourne : r0 a la place de n @ adresse de retour : lr @ temporaire r4 : adresses @ temporaire r5 : contenu </pre>	
prologue_npgcd:	<pre> mov ip, sp stmfd sp!, {fp,ip, lr, pc} sub fp, ip, #4 @ max dans la pile ==> &max : fp + 4 @ espace local : @ sauvegarde de r4 : fp - 16 @ sauvegarde de r5 : fp - 20 @ variable locale d &d : fp - 24 @ @ total 3 mots : 0xc sub sp, sp , #0xc str r4,[fp,#-16] str r5,[fp,#-20] sub r5, r1, r2 @ r5 = a - b sub r4, fp, #24 @ r4 = &d str r5, [r4] @ *&d = a - b @ 2 instructions redondantes sub r4, fp, #24 @ r4 = &d ldr r5, [r4] @ r5 = *&d str r5, [r3] @ *a_moins_b = d ldr r4, [fp, #4] @ r4 = *&max str r1, [r4] @ **&max = a cmp r2, r1 @ calcul de b-a ble b_inf_eg_a @ if (b <= a) goto ib_inf_eg_a @ on pourrait faire goto test_tq_npgcd @ instruction redondante ldr r4, [fp, #4] @ r4 = *&max str r2, [r4] @ **&max = b b_inf_eg_a: b test_tq_pgcd @ goto test corps_tq_pgcd: cmp r1, r2 @ calcul de a - b ble sinon_pgcd @ if a <= b goto sinon sub r1, r1, r2 @ a = a - b b fin_si_pgcd sinon_pgcd: sub r2, r2, r1 @ b = b - a fin_si_pgcd: @ etiquettes synonymes test_tq_pgcd: cmp r1, r2 @ comparaison redondante bne corps_tq_pgcd @ if (a<b) goto corps mul r5, r0, r1 @ r0 = f * a mov r0, r5 </pre>	

nov. 05, 18 14:13	document_td_procedure_pile.latin1.txt	Page 4/7
epilogue_pgcd:	<pre> ldr r5,[fp,#-20] ldr r4,[fp,#-16] ldmea fp, {fp, sp, pc} @ retablir fp et sp + pc <= lr </pre>	

nov. 05, 18 14:13	document_td_procedure_pile.latin1.txt	Page 5/7
Bloc param calcul -> npgcd	anc fp (de calcul)	fp_npgcd - 12
----	anc sp (de calcul)	fp_npgcd - 8
	@sse ret(ds calcul)	. fp_ngcd - 4
	@sse corps npgcd	<--- fp_ngcd
		.
sp calcul ----->	npgcd par4 (max)	<---- fp_npgcd + 4
		.
Bloc local de calcul	calcul var/tmp1 m	. fp_calc - 44
		.
	calcul sauv r9	. fp_calc - 40
		.
	calcul sauv r5	. fp_calc - 36
		.
	calcul sauv r4	. fp_calc - 32
		.
	calcul sauv r3	. fp_calc - 28
		.
	calcul sauv r2	. fp_calc - 24
		.
	calcul sauv r1 ul	. fp_calc - 20
		.
	calcul sauv r0	. fp_calc - 16
		.
Bloc param main -> calcul	ancien fp (de main)	fp_calc - 12

	ancien sp (de main)	fp_calc - 8
	@sse ret dans main	. fp_calc - 4
		.
	calcul(debut corps)	<---- fp_calc
		.
sp main ----->	calcul par4 (np)	. fp_calc + 4
		.
sp main + 4	calcul par5 (n)	. fp_calc + 8
		.
sp main + 8	calcul par6 (3max)	. fp_calc + 12
		.
Bloc local de main	main var/tmp4 t	. fp_main - 44
		.
	main var/tmp3 pg	. fp_main - 40
		.
	main var/tmp2 diff	. fp_main - 36
		.
	main var/tmp3 somme	. fp_main - 32
		.
	main sauv r7	. fp_main - 28
		.
	main sauv r4	. fp_main - 16
		.
Bloc param crt -> main	ancien fp (de crt)	fp_main - 12

	ancien sp (de crt)	. fp_main - 8
	@sse ret (dans crt)	. fp_main - 4
		.
	main (debut corps)	<---- fp_main
		.
sp crt ----->		

nov. 05, 18 14:13	document_td_procedure_pile.latin1.txt	Page 6/7
calcul:	.global calcul	
	@ procedure non terminale avec variables locales	
	@ et sauvegarde de registres	
	@	
	@ x : r0	
	@ y : r1	
	@ plus : r2	
	@ moins : r3	
	@ sp pointe sur np	
	@ sp +4 pointe sur n	
	@ sp +8 pointe sur troismax	
	@ adresse retour : lr	
prologue_calcul:		
	mov ip, sp	
	stmfd sp!, {fp, ip, lr, pc}	
	sub fp, ip, #4	
	@ n, np, troismax dans la pile	
	@ &np : fp + 4	
	@ &n : fp + 8	
	@ &troismax : fp + 12	
	@ utilisation des registres r4 a r10	
	@ calc : r9	
	@ temporaire contenus : r4	
	@ temporaire adresses : r5	
	@ espace local	
	@ parametres recus ecrases lors de l'appel a pgcd	
	@ sauvegarde de r0 (x) : fp -16	
	@ sauvegarde de r1 (y) : fp -20	
	@ sauvegarde de r2 (plus) : fp -24	
	@ sauvegarde de r3 (moins) : fp -28	
	@ registres modifiees	
	@ sauvegarde de r4 : fp -32	
	@ sauvegarde de r5 : fp -36	
	@ sauvegarde de r9 : fp -40	
	@ variable locale m : fp -44	
	sub sp, sp, #36 @ 8 locaux + 1 parametres passe	
	str r4, [fp, #-32]	
	str r5, [fp, #-36]	
	str r9, [fp, #-40]	
corps_calcul:		
	@ appel	
uire	str r0, [fp, #-16] @ l'appel a pgcd va potentiellement detruire	
	str r1, [fp, #-20] @ les contenus de r0(x), r1 (y)	
	str r2, [fp, #-24] @ r2 (plus) et r3 (moins)	
	str r3, [fp, #-28] @	INUTILE
	@ parametres d'appel	
	@ f = *n dans r0	
	@ a = y deja dans r1	
	@ b = x dans r2	
	@ a_moins_b = moins deja dans r3	
	sub r5, fp, #44 @ r5 = &m	
	str r5, [sp] @ max = &m dans pile ==> sp	

