Université de Nanterre jeudi 17 avril 2014
architecture des ordinateurs contrôle continu

```
      p1
  loadimm16 r3,1
  sub r0,r3,r0
  jc L2
L1:
  load r1,r4
  load r2,r5
  add  r4,r5,r5
  sub  r4,r5,r4
  store r1,r5
  store r2,r4
  add r1,r3,r1
  add r2,r3,r2
  sub r0,r3,r0
  jnc L1
L2:
  ret
```

```
      p2
  loadimm16 r3,1
  add r0,r1,r0
  sub r0,r3,r0
  sub r1,r0,r4
  ja L4
L3:
  load r1,r4
  load r0,r5
  neg r4,r4
  neg r5,r5
  store r1,r5
  store r0,r4
  add r1,r3,r1
  sub r0,r3,r0
  sub r1,r0,r4
  jna L3
L4:
  ret
```

```
      f
  loadimm16 r2,1
  xor r3,r3,r3
  sub r0,r2,r0
  jc L6
L5:
  load r1,r4
  mul r4,r4,r4
  mul r4,r4,r4
  mul r4,r4,r4
  add r3,r4,r3
  add r1,r2,r1
  sub r0,r2,r0
  jnc L5
L6:
  mov r3,r0
  ret
```

Retrouvez le source C des deux procédures `p1` et `p2` et de la fonction `f`.

Ecrivez l'équivalent en assembleur de la fonction C ayant pour prototype `long som10(int n, int t[]);` qui calcule $\sum_{i=0}^{n-1} t[i]^{10}$.

Dans une machine qui fait des calculs sur des entiers codés sur 4 bits, quelles valeurs auront les indicateurs CF, OF et ZF après chacune des opérations: 10+11, 10-11, 3+13, 3-13, 4+13, 4-13, 6+3, 6-3, 14+2, 14-2?

Quelle opération logique effectue le circuit dessiné au tableau ?

Corrigé

```
void p1(int n, int *t, int *u)
{ while(n--) { int x=*t, y=*u; x-=y+=x; *t++=y, *u++=x; } }
void p1(int n, int *t, int *u)
{ while(n--) { int x=*t, y=*u; *t++=x+y, *u++=-y; } }
void p2(int n, int *t)
{ int *u=t+n-1, x, y;
  while(u>t) x=*t, y=*u, *t++=-y, *u--=-x;
}
int f(int n, int *t)
{ int s=0, x;
  while(n--) x=*t++, x*=x, x*=x, s+=x*x;
  return s;
}
```

`p1(n,t,u)` prend deux tableaux d'entiers de même dimension `n` et remplace le contenu du premier par leur somme, et le contenu du second par l'opposé du premier.

`p2(n,t)` prend un tableau d'entiers de dimension `n` et remplace le premier élément du tableau et le dernier par leurs opposés échangés, puis recommence sur le sous-tableau obtenu en ignorant le premier et le dernier élément. On fait cela tant qu'il y a plusieurs éléments dans le tableau. Si le tableau est de taille impaire, son élément central sera inchangé. A part cela le tableau est retourné et opposé.

$$\texttt{f}(\texttt{n},\texttt{t}) = \sum_{i=0}^{n-1} t[i]^8$$

```
                    // int som10(int n, int *t)
  loadimm16 r2,1    // {int r2=1;
  xor r3,r3,r3      //  int s=0;
  sub r0,r2,r0      //  if(n--)
  jc L2             //  do
L1:                 //  {
  load r1,r4        //    int x=*t;
  mul r4,r4,r5      //    int y=x*x;  // *t^2
  mul r5,r5,r4      //    x=y*y;      // *t^4
  mul r4,r4,r4      //    x*=x;       // *t^8
  mul r4,r5,r4      //    x*=y;       // *t^10
  add r4,r3,r3      //    s+=x;
  add r1,r2,r1      //    t++;
  sub r0,r2,r0      //  } while(n--);
  jnc L1
L2:
  mov r3,r0         //  return s;
  ret               // }
```

non signé	CF	signé	OF	ZF	SF
10+11=5	1	-6+-5=5	1	0	0
10-11=15	1	-6--5=-1	0	0	1
3+13=0	1	3+-3=0	0	1	0
3-13=6	1	3--3=6	0	0	0
4+13=1	1	4+-3=1	0	0	0
4-13=7	1	4--3=7	0	0	0
6+3=9	0	6+3=-7	1	0	1
6-3=3	0	6-3=3	0	0	0
14+2=0	1	-2+2=0	0	1	0
14-2=12	0	-2-2=-4	0	0	1

Le circuit calcule $\bar{A} \wedge \bar{B} \vee A \wedge \bar{C}$ ou `A?!C:!B`

Barème

p1	3pt	2.5pt pour du vrai C ou 1pt si r4=*r1 ...
		0.5pt pour une explication en français
p2	3pt	idem
f	3pt	idem
$\sum_{i=0}^{n-1} t[i]^{10}$	3pt	1pt boucle
		0.5pt valeur de retour
		1.5pt x^{10} en 4 instructions
		ou 1pt si juste mais plus de 4 instr. ou 0.5pt si idée
10+11	7.5pt	=10x0.75pt
		Chaque opération est notée comme un QCM indépendant
		avec une note entre 0 et 0.75pt.
		CF juste:0.25 absent:0 faux:-0.25
		OF:0.25, ZF:0.125, SF:0.125
A?!B:!C	2pt	
	21.5pt	