

Exercice 1 : pgcd : algorithme d'Euclide

version signée

pgcd:	pgcd:	int pgcd(int a, int b) { int c;
and r1,r1,r1	7 1 1 1	// r0 r1 r2
je fin:	34 0 0 6	while(b!=0)
sarimm r0,31,r2	15 0 31 2	{ c=a>>31; // =a<0? -1 :0;
idiv r0,r2,r1	16 0 2 1	c=a/b, a%=b;
mov r0,r2	60 0 2 *	c=a; // a<->b
mov r1,r0	60 1 0 *	a=b;
mov r2,r1	60 2 1 *	b=c;
jmp pgcd	44 255 255 248	}
fin:		
ret	29 * * *	}

version non signée

pgcd:	pgcd:	typedef unsigned uint;
xor r2,r2,r2	8 2 2 2	int pgcd(int a, int b)
sub r2,r0,r2	2 2 0 2	{ int c=0;
movcg r2,r0	54 2 0 *	c=0-a;
xor r2,r2,r2	8 2 2 2	if(0>a) a=c; // a= a
sub r2,r1,r2	2 2 1 2	c=0;
movcg r2,r1	54 2 1 *	c=0-b;
// jmp pgcd		if(0>b) b=c; // b= b
		return pgcd(a,b);
		}
pgcd:	pgcd:	uint pgcd(uint a, uint b) { uint c;
and r1,r1,r1	7 1 1 1	// r0 r1 r2
je fin:	34 0 0 6	while(b!=0)
xor r2,r2,r2	8 2 2 2	{ c=0;
div r0,r2,r1	17 0 2 1	c=a/b, a%=b;
mov r0,r2	60 0 2 *	c=a; // a<->b
mov r1,r0	60 1 0 *	a=b;
mov r2,r1	60 2 1 *	b=c;
jmp pgcd	44 255 255 248	}
fin:		
ret	29 * * *	}

Exercice 2 : pgcd sans division : algorithme binaire

```

//      r0      r1      r2      r3
unsigned pgcdb(unsigned a,unsigned b)//  d      j
{ unsigned d=a|b,          or  r0,r1,r2      6  0  1  2
  j=__builtin_ctz(d);      bsf  r2,r3      30  2  3  *
  d=__builtin_ctz(a);      bsf  r0,r2      30  0  2  *
  a>>=d;                   shr  r0,r2,r0     10  0  2  0
  d=__builtin_ctz(b);      bsf  r1,r2      30  1  2  *
  b>>=d;                   shr  r1,r2,r1     10  1  2  1
  if(a==0)                 and  r0,r0,r0     7   0  0  0
    goto fin;              je  fin      34  0  0  11
  if(b==0)                 and  r1,r1,r1     7   1  1  1
    goto fin;              je  fin      34  0  0  9
  debut:                   debut:
  d=a-b;                   sub  r0,r1,r2     2   0  1  2
  if(a==b) goto fin;       je  fin      34  0  0  7
  if(a<b) d=a,             movcb r0,r2     48  0  2  *
    a=b,                   movcb r1,r0     48  1  0  *
    b=d;                   movcb r2,r1     48  2  1  *
  a-=b;                   sub  r0,r1,r0     2   0  1  0
  d=__builtin_ctz(a);      bsf  r0,r2      30  0  2  *
  a>>=d;                   shr  r0,r2,r0     10  0  2  0
  goto debut;              jmp  debut     44 255 255 245
  fin:                     fin:
  a|=b;                   or   r0,r1,r0     6   0  1  0
  a<=<=j;                  shl  r0,r3,r0     12  0  3  0
  return a;                ret              29  *  *  *
}

```

Exercice 3 : fractions

```

typedef struct{ int num,den; } frac;
//r0      r1      r2      r3      r4      r5      r6      r7      r8      r9      r10      r11      r12      r13      r14
//a.num a.den b.num .den e.num .den c.num .den d.num .den s.num .den i      imax 1
frac f()      f:
{ int un=1, i=10,imax=1000; loadimm16 r14,1; loadimm16 r12,10; loadimm16 r13,1000
  frac a,s={0,1};      xor  r10,r10,r10; mov  r14,r11
  do                  l20:
  { a.den=i*i*i-i;      mul  r12,r12,r1; mul  r1,r12,r1; sub  r1,r12,r1
    i++;                add  r12,r14,r12
    a.num=i+1;          add  r12,r14,r0
    a=reduit(a);        call  reduit
    frac b=s;           mov  r10,r2; mov  r11,r3
    a=addfrac(a,b);     call  addfrac
    s=a;               mov  r0,r10; mov  r1,r11
  } while(i<=imax);    sub  r12,r13,r2; jle  l20
  return a;            ret
}

```

Exercice 4 :

```

p:                                void p(int*u, int*v, int n) // 1 a b x,v
loadimm16 r3,1                    { //      r0      r1      r2      r3 r4 r5 r6
l1:
sub r2,r3,r6                      2  2 3 6          while(n--)
jc l2                             32 0 0 12         {
load r0,r4                       24 0 4 *          int a=*u,
load r1,r5                       24 1 5 *          b=*v;
xor r4,r5,r4                      8  4 5 4          a^=b;
sub r4,r5,r6                      2  4 5 6          int x=a-b, *w;
movcl r0,r6                      57 0 6 *          if(a<b) w=u,
movcl r1,r0                      57 1 0 *          u=v,
movcl r6,r1                      57 6 1 *          v=w;
store r0,r4                      25 0 4 *          *u=a;
store r1,r5                      25 1 5 *          *v=b;
add r0,r3,r0                     0  0 3 0          u++;
sub r1,r3,r1                     2  1 3 1          v--;
jmp l1                          44 255 255 242    }
l2:
ret                               29              }

void p(int*u, int*v, int n)
{ while(n--)
  { int a=*u, b=*v, *w;
    a^=b;
    if(a<b) w=u, u=v, v=w; // On échange u et v.
    *u++=a; *v--=b;
  } }

int t[]={1,2,8, 4, 5, 6, 7, 8, 9,10}; p(t+3,t+6,4); p(t+4,t+5,5);
*t      t[3]      t[6]      t[9]  u   v   n   a   b
{1,2,8, 4, 5, 6, 7, 8, 9,10}  t+3 t+6  3  4  7          a=t[3]=4 b=t[6]=7
                                t+6 t+3    3< 7    4^7=3 t[6]=3 t[3]=7
{1,2,8, 7, 5, 6, 3, 8, 9,10}  t+7 t+2  2  8  8          a=t[7]=8 b=t[2]=8
                                t+2 t+7    0< 8    8^8=0 t[2]=0 t[7]=8
{1,2,0, 7, 5, 6, 3, 8, 9,10}  t+3 t+6  1  7  3          a=t[3]=7 b=t[6]=3
                                t+3 t+6    4  3    7^3=4 t[3]=4 t[6]=3
{1,2,0, 4, 5, 6, 3, 8, 9,10}  t+4 t+5  0  5  6          a=t[4]=5 b=t[5]=6
                                t+5 t+4    3< 6    5^6=3 t[5]=3 t[4]=6
{1,2,0, 4, 6, 3, 3, 8, 9,10}  t+4 t+5  4  6  3          a=t[4]=6 b=t[5]=3
                                t+4 t+5    5  3    6^3=5 t[4]=5 t[5]=3
{1,2,0, 4, 5, 3, 3, 8, 9,10}  t+5 t+4  3  3  5          a=t[5]=3 b=t[4]=5
                                t+5 t+4    6  5    3^5=6 t[5]=6 t[4]=5
{1,2,0, 4, 5, 6, 3, 8, 9,10}  t+6 t+3  2  3  4          a=t[6]=3 b=t[3]=4
                                t+6 t+3    7  4    3^5=6 t[6]=7 t[3]=4
{1,2,0, 4, 5, 6, 7, 8, 9,10}  t+7 t+2  1  8  0          a=t[7]=8 b=t[2]=0
                                t+7 t+2    8  0    8^0=8 t[7]=8 t[2]=0
{1,2,0, 4, 5, 6, 7, 8, 9,10}  t+8 t+1  0  9  2          a=t[8]=9 b=t[1]=2
                                t+8 t+1   11  2    9^2=11 t[8]=11 t[1]=2
{1,2,0, 4, 5, 6, 7, 8,11,10}

```