```
> #Correction TP 2
> #I. Exos d'arithmétique
  #Exo 1:
> nombrededeux:=proc(n)
  local quotient, reste, nombrede2;
  quotient:=n;
  nombrede2:=0;
  reste:=irem(quotient,2);
  while (reste = 0) do
  nombrede2 := nombrede2 +1;
  quotient:=iquo(quotient,2);
  reste:=irem(quotient,2);
  end do;
  return nombrede2;
  end proc;
```

nombrededeux := **proc**(*n*) (1)

```
    local quotient, reste, nombrede2;
    quotient := n;
    nombrede2 := 0;
    reste := irem(quotient, 2);
    while reste = 0 do
        nombrede2 := nombrede2 + 1;
        quotient := iquo(quotient, 2);
        reste := irem(quotient, 2)
    end do;
    return nombrede2
end proc
```

```
> nombrededeux(155); ifactor(155);
```

0 (2)

(5) (31)

```
> nombrededeux(4640); ifactor(4640);
```

5 (3)

$(2)^5$ (5) (29)

```
> #EXO 2 :
> #Q1 :
> densitePremiers := proc(n)
  local nb, i,d;
  nb:=1;
  for i from 3 to n do
  if isprime (i) then
  nb:=nb+1;
  end if;
  end do;
  d:=evalf(nb/n);
```

```
    return d;
    end proc;
densitePremiers := proc(n)                                              (4)
    local nb, i, d;
    nb := 1;
    for i from 3 to n do
        if isprime(i) then
            nb := nb + 1
        end if
    end do;
    d := evalf(nb/n);
    return d
end proc
> densitePremiers (10), densitePremiers(100), densitePremiers
    (1000000);
                    0.4000000000, 0.2500000000, 0.07849800000           (5)
> #Q. 4 :
> densiteJumaux := proc (n)
    local nb, i, d;
    nb := 0;
    for i from 3 to n do
    if isprime(i) and isprime(i+2) and i+2 <= n then
    nb:=nb+1;
    end if;
    end do;
    d:=evalf(nb/n);
    return d;
    end proc;
densiteJumaux := proc(n)                                                (6)
    local nb, i, d;
    nb := 0;
    for i from 3 to n do
        if isprime(i) and isprime(i + 2) and i + 2 <= n then
            nb := nb + 1
        end if
    end do;
    d := evalf(nb/n);
    return d
end proc
> densiteJumaux(6);
                              0.1666666667                              (7)
> #EX0 3:
    #Q. 1
> image := proc(f,m,n)
    local L, i;
    L:=[];
```

```
   for i from m to n do
   L:=[op(L),f(i)];
   end do;
   return L;
   end proc;
```

```
image := proc(f, m, n)
    local L, i;
    L := [ ];
    for i from m to n do
        L := [op(L), f(i)]
    end do;
    return L
end proc
```

(8)

```
> #Q. 2 :
> p:=x->x^2+x+41;
```

$$p := x \to x^2 + x + 41 \tag{9}$$

```
> L:=image(p,-40,40);
```

$$L := [1601, 1523, 1447, 1373, 1301, 1231, 1163, 1097, 1033, 971, 911, 853, 797, 743, 691, 641, 593, 547, 503, 461, 421, 383, 347, 313, 281, 251, 223, 197, 173, 151, 131, 113, 97, 83, 71, 61, 53, 47, 43, 41, 41, 43, 47, 53, 61, 71, 83, 97, 113, 131, 151, 173, 197, 223, 251, 281, 313, 347, 383, 421, 461, 503, 547, 593, 641, 691, 743, 797, 853, 911, 971, 1033, 1097, 1163, 1231, 1301, 1373, 1447, 1523, 1601, 1681] \tag{10}$$

```
> #Q. 3 :
> premiers := proc(L)
   local M,i;
   M:=[];
   for i from 1 to nops(L) do
   if isprime(L[i]) then
   M:=[op(M),L[i]];
   end if;
   end do;
   return M;
   end proc;
```

```
premiers := proc(L)
    local M, i;
    M := [ ];
    for i to nops(L) do
        if isprime(L[i]) then
            M := [op(M), L[i]]
        end if
    end do;
    return M
end proc
```

(11)

```
> #Q. 4 :
> M:=premiers(L);
>
```

```
M := [1601, 1523, 1447, 1373, 1301, 1231, 1163, 1097, 1033, 971, 911, 853, 797, 743, 691, 641,
    593, 547, 503, 461, 421, 383, 347, 313, 281, 251, 223, 197, 173, 151, 131, 113, 97, 83, 71, 61,
    53, 47, 43, 41, 41, 43, 47, 53, 61, 71, 83, 97, 113, 131, 151, 173, 197, 223, 251, 281, 313, 347,
    383, 421, 461, 503, 547, 593, 641, 691, 743, 797, 853, 911, 971, 1033, 1097, 1163, 1231,
    1301, 1373, 1447, 1523, 1601 ]                                                              (12)
> #Constation :
> nops(L), nops(M);
                                        81, 80                                                  (13)
> #Sur les 81 images obtenues, 80 sont des nombres premiers!


> #EXO 4 :
> restart;
> fermat:=proc()
  local k, F;
  k:=0;
  F:=2^(2^k)+1;
  while isprime(F) do
  printf("F%d=%d\n",k,F);
  k:=k+1;
  F:=2^(2^k)+1;
  end do;
  printf("Le %d eme nombre de Fermat n'est pas premier.\n",k);
  printf("F%d=%d=%A",k,F,ifactor(F));
  end proc;
fermat := proc( )                                                                               (14)
    local k, F;
    k := 0;
    F := 2^(2^k) + 1;
    while isprime(F) do
        printf("F%d=%d
            ", k, F);
        k := k + 1;
        F := 2^(2^k) + 1
    end do;
    printf("Le %d eme nombre de Fermat n'est pas premier.
        ", k);
    printf("F%d=%d=%A", k, F, ifactor(F))
end proc
> fermat();
F0=3
F1=5
F2=17
F3=257
F4=65537
Le 5 eme nombre de Fermat n'est pas premier.
```

F5=4294967297=(641)*(6700417)

```
> #Exo 5 :

> # Q1

> somme := proc(n)
  local s, m;
  s := 0;
  m := n;
  while(m <> 0) do
  s := s + irem(m, 10);
  m := iquo(m, 10);
  end do;
  return s;
  end proc;

> # Q3
  itereSomme := proc(n)
  local m, s;
  m := n;
  s := 0;
  while (m >= 10) do
  s := somme(m);
  m := s;
  end do;
  return s;
  end proc;
```

```
itereSomme := proc(n)                                          (15)
    local m, s;
    m := n;
    s := 0;
    while 10 <= m do
        s := somme(m);
        m := s
    end do;
    return s
end proc
```

```
> #Exo 6 :
  restart;

> facteurs := proc (n)
  local L,i,m;
  L:=[];
  i:=2;
```

```
   m:=n;
   while m<>1 do
   if irem(m,i)=0 then
   m:=iquo(m,i);
   L:=[op(L),i];
   else
   i:=i+1;
   end if;
   end do;
   return L;
   end proc;
```

```
facteurs := proc(n)
    local L, i, m;
    L := [ ];
    i := 2;
    m := n;
    while m <> 1 do
        if irem(m, i) = 0 then
            m := iquo(m, i);
            L := [op(L), i]
        else
            i := i + 1
        end if
    end do;
    return L
end proc
```

(16)

```
> facteurs (360);
```

$$[2, 2, 2, 3, 3, 5] \tag{17}$$

```
> facteurs (81);
```

$$[3, 3, 3, 3] \tag{18}$$

```
> facteurs (13);
```

$$[13] \tag{19}$$

```
> facteurs(1);
```

$$[\,] \tag{20}$$

```
> #Exo 7 :
>
> restart;
> pgcd := proc(a,b)
   local u,v,r;
   u:=a;
   v:=b;
   r:=1;
   while r > 0 do
   r:=irem(u,v);
   u:=v;
```

```
    v:=r;
    end do;
    return u;
    end proc;
```

$$pgcd := \mathbf{proc}(a, b) \tag{21}$$

```
    local u, v, r;
    u := a;
    v := b;
    r := 1;
    while 0 < r do
        r := irem(u, v);
        u := v;
        v := r
    end do;
    return u
end proc
```

```
> pgcd(16,15), igcd(16,15);
```

$$1, 1 \tag{22}$$

```
> pgcd(48,4),igcd(48,4);
```

$$4, 4 \tag{23}$$

```
> pgcd(4,48),igcd(4,48);
```

$$4, 4 \tag{24}$$

```
> pgcd(123456789,987654321),igcd(123456789,987654321);
```

$$9, 9 \tag{25}$$

```
> pgcd(123456789987654321,15975325845695175385265 4),igcd
  (123456789987654321,15975325845695175385265 4);
```

$$33, 33 \tag{26}$$

```
>
```