

#exo1

> premiers:=proc(N)

local S, k;

S:=NULL;

for k to N do

if isprime(k) then S:=S, k fi

od;

S;

end;

> premiers(70);

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67

> premiers(100);

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

> #exo2

> #Écrivez une procédure diviseurs qui renvoie la liste des diviseurs d'un entier n.

> Diviseurs := proc(n)

local i, diviseurs;

diviseurs := [];

for i from 1 to n do

if n mod i = 0 then

diviseurs := [op(diviseurs), i];

end if;

end do;

return diviseurs;

```

end proc;
> Diviseurs(12);

      [1, 2, 3, 4, 6, 12]

> restart;

> with(numtheory):

> mersenne_numbers := proc(n)

    local p, M;

    for p from 2 to n do

        if isprime(p) then

            M := 2^p - 1;

            print(M);

        end if;

    end do;

end proc;

> mersenne_numbers(2^3 - 1);

      3

      7

     31

    127

```

#Exo 3

```

> Euclide := proc(a, b)

    local u, v, r;

    u := a:

    v := b:

    while v <> 0 do

```

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

```
> Euclide(252, 105);
```