(\* TP2 \*)

(\* Nombres premiers \*)

(\* Q1 \*)

let rec less\_divider (i:int) (n:int) : int =

if ( ( ( n mod i ) == 0 ) && ( n > i )) then i

else if (i>=n) then 0

else less\_divider (i+1)(n)

(\* Q2 \*)

let prime (n:int) : bool =

if (n<=1) then false

else if ((less\_divider 2 n)=0) then true

else false

(\* Q3 \*)

let rec next\_prime (n:int) : int =

if ( (prime n) = true) then n

else next\_prime (n+1)

(\* Q4 \*)

let rec nth\_prime (n:int) : int =

if (n<0) then failwith "invalid exponent"

else if n=0 then 2

else next\_prime (nth\_prime (n-1) +1)

(\* Approximation de la racine carrée \*)

let r0 = 1.0

(\* Q5 \*)

let r (a:float) (x:float) : float =

1.0/.2.0\*.(x+.(a/.x))

(\* Q6 \*)

let rec sqrt\_n (n:int) (a:float) =

if (n=0) then r0

else (r a (sqrt\_n (n-1) a))

(\* Q7 \*)

let eq\_eps (e:float) (x:float) (y:float) : bool =

abs\_float(x-.y) <=e

(\* Q8 \*)

let sqrt\_x (e:float) (a:float) : float =

let rec loop (s:float) (s':float) : float =

if eq\_eps e s s' then s

else loop s' (r a s')

in

loop r0 (r a r0)