**RECUIT SIMULÉ**

Question 1)

La commande **init n max** renvoie un couple villes, circuit.

**villes** correspond à un vecteur de taille **n** dont l’élément d’indice i est un vecteur contenant les coordonnées de la ville **i**, celles-ci appartenant [0, max] x [0, max].

**circuit** correspond à un premier parcours des villes dans l’ordre croissant de leurs indices.

Question 2)

let distance v1 v2 =

sqrt( ( float\_of\_int(v2.(0) - v1.(0)) )\*\*.2.

+. ( float\_of\_int(v2.(1) - v1.(1)) )\*\*.2. ) ;;

Question 3)

let longueur\_circuit circuit villes =

let l\_c = ref(0.) in

let n = vect\_length villes in

for i = 0 to n-2 do

l\_c := !l\_c +. (distance villes.(circuit.(i)) villes.(circuit.(i+1))) ;

done;

l\_c := !l\_c +. (distance villes.(circuit.(n-1)) villes.(circuit.(0))) ;

!l\_c ;;

Question 4)

let affiche\_circuit circuit villes =

let max\_coord c v u =

let m = ref(v.(0).(u)) in

for i = 1 to ((vect\_length c) - 1) do

m := max v.(i-1).(u) v.(i).(u) ;

done;

!m

in

let n = vect\_length villes in

let max\_x = ref(max\_coord circuit villes 0) in

let max\_y = ref(max\_coord circuit villes 1) in

open\_graph ((string\_of\_int(!max\_x + 10))^("\*"^((string\_of\_int(!max\_y + 10))^"0-0")));

set\_color red;

fill\_circle villes.(circuit.(0)).(0) villes.(circuit.(0)).(1) 5;

for i = 1 to (n - 1) do

set\_color blue;

moveto villes.(circuit.(i-1)).(0) villes.(circuit.(i-1)).(1);

lineto villes.(circuit.(i)).(0) villes.(circuit.(i)).(1);

set\_color red;

fill\_circle villes.(circuit.(i)).(0) villes.(circuit.(i)).(1) 5;

done;

set\_color blue;

moveto villes.(circuit.(n-1)).(0) villes.(circuit.(n-1)).(1);

lineto villes.(circuit.(0)).(0) villes.(circuit.(0)).(1);

set\_color red;

fill\_circle villes.(circuit.(0)).(0) villes.(circuit.(0)).(1) 5;;

Question 5)

let recuit circuit villes delta n\_r =

let swap c i j =

let cp = ref(c) in

let x = !cp.(i) in

!cp.(i) <- !cp.(j);

!cp.(j) <- x ;

!cp

in

let d c1 c2 =

(longueur\_circuit c2 villes) -. (longueur\_circuit c1 villes)

in

let c1 = ref(circuit) in

let k = ref(0.99) in

let n = vect\_length circuit in

let delta\_ref = ref(delta) in

for i = 0 to n\_r - 1 do

let c2 = swap !c1 (random\_\_int n) (random\_\_int n) in

if d !c1 c2 < 0. then c1 := c2

else

let r = random\_\_float 1. in

let e = exp(-.(d !c1 c2)/.(!delta\_ref)) in

if e > r then c1 := c2 ;

delta\_ref := !delta\_ref \*. !k

done;

!c1;;

Question 6)

let recuit\_simule circuit villes delta n\_r n\_e =

let c = ref(circuit) in

let delta\_r = ref(delta) in

for i = 0 to n\_e - 1 do

let c\_tp = recuit !c villes !delta\_r n\_r in

c := c\_tp ;

delta\_r := random\_\_float(5.) ;

done;

!c ;;

let v, c = init 15 300;;

affiche\_circuit c v;;

longueur\_circuit c v;;

let c1 = recuit\_simule c v 1. 5000 10 ;;

affiche\_circuit c1 v;;

longueur\_circuit c1 v;;