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;;
```