Exam programmation fonctionnelle

1. Écrivez une fonction "distance p1 p2" qui renvoie la distance entre p1 et p2. On rappelle que si (x1,y1) et (x2,y2) sont les coordonnées de p1 et p2, la distance vaut la racine carrée (sqrt) de (x2-x1)^2 + (y2-y1)^2.
   Par exemple, distance (1.,1.) (2.,2.) envoie 1.41421356237309515 (racine carrée de 2).

2. On considère le type suivant (déjà défini dans le fichier).
   type date = {day:int; month:int; year:int}.
   Écrivez une fonction "date_to_string date" qui renvoie la chaîne de caractères qui représente la date au format day/month/year.
   Par exemple, date_to_string {day=10;month=5;year=2013} renvoie "10/5/2013”

3. Écrivez une fonction un_sur_deux t qui renvoie un tableau t2 de longueur moitié de t contenant les éléments aux indices pairs de t (autrement dit, un élément sur deux).
   Par exemple, un_sur_deux [|0;10;7;5;4|] renvoie [|0;7;4|].

4. Écrivez une fonction moyenne2_list l qui renvoie une liste l2 de flottants de longueur moitié contenant les moyennes des paires d'éléments successifs: le premier élément de l2 est la moyenne des deux premiers éléments de l, le deuxième élément de l2 est la moyenne des deux suivants, etc.
   Si la liste est de longueur impaire, la dernière moyenne est calculée sur un seul élément.
   Par exemple, moyenne2_list [1.0;2.0;3.0;4.0;5.0] renvoie [1.5;3.5;5.0]

5. La suite de Syracuse ($u_{i>=0}$) d'ordre n est définie par u_0=n et $u_{i+1} = u_i/2$ si u_i est pair, sinon $u_{i+1}=3u_i+1$. La suite se termine lorsque u_i=1.
   Par exemple, pour n=12, la suite de Syracuse d'ordre 12 est 12,6,3,10,5,16,8,4,2,1.
   L'altitude maximale de Syracuse(n) est le plus grand entier de la suite de Syracuse d'ordre n.
   Écrivez une fonction altitude_maximale n qui renvoie l'altitude maximale de Syracuse(n).
   Par exemple, altitude_maximale 12 renvoie 16.

6. Une sous-suite d'une liste l=[a_1;a_2;a_3;...;a_n] est une liste [a_i;a_{i+1};a_{i+2};...;a_j]. Cette sous-suite est croissante si $a_i <= a_{i+1} <= a_{i+2} <= ... <= a_j$.
   Écrivez une fonction plus_longue_sous_suite_croissante l qui renvoie la longueur de la (ou les) plus longue(s) sous-suite(s) croissante(s) de l.
   Par exemple, plus_longue_sous_suite_croissante [1;3;0;2;5;1] renvoie 3 car la plus longue sous-suite croissante, [0;2;5], est de longueur 3.

7. On considère le type ci-dessous (prédéfini dans le fichier)
   type id = {name:string; surname:string; age:int}
   Écrivez une fonction sort_by_name l qui prend une liste d'ids et qui renvoie cette liste triée par rapport au champ name.

8. On considère le type suivant pour les arbres binaires d'expression
   type arbre =Const of int | Var of string | Plus of arbre * arbre
   Écrivez une fonction nb_feuilles arbre qui renvoie le nombre de feuilles (Const ou Var) de l'arbre binaire d'expression.
   Par exemple, nb_feuilles (Plus(Const 1, Var "x")) renvoie 2.

9. Écrivez une fonction last_line ic qui prend en argument un in_channel ic (~ un descripteur de fichier) et renvoie la dernière ligne du fichier. Si le fichier est vide, la fonction renvoie la chaîne de caractères vide.
   Indication : On rappelle que input_line ic renvoie une ligne du fichier et passe à la ligne suivante pour le prochain appel à input_line. Cette fonction lève l'exception End_of_file quand la fin du fichier est atteinte.
   Par exemple, si le fichier "toto.txt" contient
   hello
   world
   on aura last_line (open_in "toto.ml") qui renvoie "world"

10. Un couple (x,y) représente une tuile de domino. Une suite de tuiles est correcte si le deuxième élément d'une tuile correspond au premier élément de la tuile suivante.
    Par exemple, la suite [(1,2);(2,5);(5,3)] est correcte.
    Écrivez une fonction domino l qui prend en argument une liste de tuiles et qui renvoie true s'il est possible, en réordonnant les tuiles et en permutant éventuellement les deux côtés d'une tuile, de faire une suite correcte.
    Par exemple, domino [(1,2);(3,5);(2,5)] renvoie true car on peut former la suite correcte de l'exemple précédent.