

TD de révisions

Fonctions en Racket, listes, récursivité.

Question 1. Définir une fonction *fct* calculant la valeur de $f(x) = \frac{1}{1 + (x - 2)^2}$ sous forme d'un nombre flottant.

(fct 1) renvoie 0.5.

Question 2. Définir une fonction *f* prenant en paramètres un entier n , et retournant la valeur de la suite $f(n)$ où $f(n)$ est définie par

$$f(0) = 3, \quad \text{et pour } n \geq 1, f(n) = 1 + 2 \times f(n - 1).$$

(f 8) renvoie 1023.

Question 3. Définir une fonction *nuple* prenant en paramètres un entier n et renvoyant la liste de toutes les valeurs $f(i)$ pour $0 \leq i \leq n$.

(nuple 5) renvoie '(127 63 31 15 7 3).

Question 4. Définir une fonction *binaire* prend en paramètres un nombre x et renvoie une liste correspondant à l'écriture binaire de x . On pourra utiliser la fonction *quotient* pour obtenir le quotient de la division de x par 2.

(binaire 42) renvoie '(1 0 1 0 1 0).

Question 5. Définir une fonction *firsts* retournant les éléments d'une liste sans le dernier.

(firsts '(1 2 3 4 5)) renvoie '(1 2 3 4).

(firsts '(5 4 3 2 1)) retourne '(5 4 3 2).

Question 6. Définir une fonction *appartient* qui prend en paramètres une liste L et un entier x , et renvoie un booléen spécifiant si x appartient à L ou non. Utiliser cette fonction pour définir une fonction *doublon* qui renvoie une liste L sans doublons.

(doublon '(1 2 3 1 2)) renvoie '(3 1 2).

Question 7. Définir une fonction *nbsupx* qui, étant donnés une liste de nombres et un nombre x , compte le nombre d'éléments de L strictement inférieurs à x .

(nbsupx '(7 4 9 2 6 10 8) 9) renvoie 5.

Question 8. Définir une fonction *repete* qui renvoie une liste dont tous les éléments sont répétés deux fois à la suite.

(repete '(1 2 3 4) renvoie '(1 1 2 2 3 3 4 4).

Question 9. Définir une fonction *regroupe* qui prend en paramètre une liste de longueur paire (on ne vérifiera pas que c'est le cas), et regroupe deux éléments consécutifs dans une liste. Si la liste a un nombre impair d'éléments, on laissera le dernier tout seul.

(regroupe '(a b c d e f)) renvoie '((a b) (c d) (e f))

(regroupe '(a b c d e f g) renvoie '((a b) (c d) (e f) (g)).

Question 10. Définir une fonction *remplace* qui prend en paramètres une liste L , et deux symboles u et v , et qui remplace toutes les occurrences de u par v dans L .

(remplace '(b o n j o u r) 'o 'i) renvoie '(b i n j i u r).

Question 11. Définir une fonction *separe* qui, étant donné une liste, renvoie une liste de deux sous-listes : celle contenant les symboles qui ne sont pas des nombres et celle contenant les nombres.

(separe '("a" 1 "b" "c" 2 3 "d" 4)) renvoie (("a" "b" "c" "d") (1 2 3 4))

Question 12. Définir une fonction *max-occ* qui calcule le maximum d'une liste de nombres ainsi que le nombre d'occurrences de ce maximum.

(max-occ '(3 6 2 3 6 4 1 6)) renvoie '(6 3).

Question 13. Définir une fonction *nfois* qui prend en paramètres une liste L , un élément e et un entier n , et qui renvoie $\#t$ si e apparaît n fois dans L , $\#f$ sinon.

(nfois '(1 2 3 4 4) 5 0) renvoie $\#t$.

(nfois '(1 2 3 4 4) 5 1) renvoie $\#t$.