Pôle Universitaire Léonard de Vinci
ESI CS 2506 – Applications du Langage C

S5 – Année 2009-2010
Promotion ESILV 2012

Feuille d'exercices 3

Applications

Exercice I

Soit f la fonction de $\mathbb{N}^*$ dans $\mathbb{N}^*$ définie par

$$f(n) = \begin{cases} n/2 & \text{si} \quad n \text{ est pair} \\ 3n+1 & \text{si} \quad n \text{ est impair} \end{cases}$$

Les suites (u_n) définies par $u_{n+1} = f(u_n)$ s'appellent suites de Syracuse. La conjecture de Collatz[1] dit que

$$\forall u_0 \in \mathbb{N}^*,\ \exists N \in \mathbb{N}^*,\ u_N = 1$$

Vérifier cette conjecture pour u_0 parmis les 1000 premiers entiers.

Exercice II

On se donne une fonction continue g (dans le programme vous pourrez tester avec $g(x) = x^2$) et deux réels a et b (dans le programme vous pourrez tester avec $a = 0$ et $b = 5$). On souhaite donner une valeur approchée de $\int_a^b g(x)\,dx$ Visionner la séquence vidéo sur l'integration à http://tinyurl.com/nw9cfc et implémenter les quatre premiers algorithmes proposés.

Exercice III

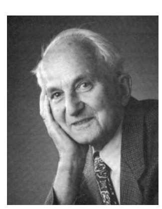
L'objet de cet exercice est de mettre en place un type nombres rationnels `Q` afin de pouvoir faire des opérations exactes sur ces nombres.

1. Un nombre rationnel sera stocké sous la forme d'un type structuré avec deux champs : `N` pour le numérateur et `D` pour le dénominateur. Donnez les instructions permettant de créer ce type.

2. Faire les fonctions suivantes

- `creerQ` qui prend comme argument deux entiers et retourne le nombre rationnel correspondant.
- `afficherQ` qui prend comme argument un nombre rationnel et l'affiche à l'écran (l'affichage peut se faire sur une ligne ou sur trois, à votre convenence).

[1]Lothar Collatz, mathématicien allemand du XXe siècle, conjectura en 1937 que toute suite de Syracuse atteignait la valeur 1. Il contribua également de manière importante à l'analyse numérique et aux équations aux dérivées partielles.



Lothar Collatz (1910–1990)

- `approximerQ` qui prend comme argument un nombre rationnel et retourne un réel double précision l'approximant.
- `egalQ` qui prend comme arguments deux nombres rationnels et retourne 1 s'ils sont égaux et 0 sinon.
- `produitQ` qui prend comme arguments deux nombres rationnels et retourne leur produit.
- `divisionQ` qui prend comme arguments deux nombres rationnels, le deuxième étant non nul, et retourne leur quotient.
- `sommeQ` qui prend comme arguments deux nombres rationnels et retourne leur somme.
- `irreductibleQ` qui prend comme arguments un nombre rationnels et retourne un nombre rationnel égal sous forme irréductible.

3. Mettre les types et les fonctions dans une bibliothèque. Faire un programme qui affiche la fraction irréductible correspondant à $(1/2 + 1/4)/(2/9 + 1/7)$ et une valeur approchée.

Exercice IV

On lance une pièce de monnaie 10 fois de suite. Réaliser un programme qui estime la probabilité d'obtenir 3 faces exactement. Comparer avec les résultats trouvés l'année dernière en MF 1302.

Exercice V

On considère l'ensemble des fonctions homographiques

$$H = \left\{ \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \frac{ax+b}{cx+d} \end{array} \quad , \quad (a,b,c,d) \in \mathbb{R}^4 \right\}$$

1. Faire un type `H` pour les fonctions homographiques.

2. Faire les fonctions suivantes :

- `CreerH` qui prend comme argument quatre réels et retourne la fonction homographique associée.
- `EstAffineH` qui prend comme argument une fonction homographique et retourne 1 si la fonction est affine et 0 sinon.
- `PointSingulierH` qui prend comme argument une fonction homographique non-affine et qui retourne le réel où elle n'est pas défini.
- `CalculerH` qui prend comme argument une fonction homographique f et un réel x auquel elle est définie et qui retourne $f(x)$.
- `EstEgalH` qui prend comme argument deux fonctions homographiques et qui retourne 1 si ces fonctions sont égales et 0 sinon.

3. Soit f une fonction homographique non constante, montrer que la fonction réciproque f^{-1} est également homographique. Faire une fonction `InverseH` qui prend comme argument une fonction homographique f et qui retourne f^{-1}.

4. La composée de 2 fonctions homographiques est-elle toujours une fonction homographique ? Si oui, faire une fonction `ComposeH` qui prend comme argument deux fonctions homographiques f et g et qui retourne $f \circ g$.

5. Faire un programme illustrant l'utilisation de chacune des fonction qui a pu être réalisée aux questions précédentes.