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 $a = 1$ et $b = 2$). Donnez une valeur approchée de

$$\int_a^b g(x)\,dx$$

au moyen de la méthode des trapèzes,

Exercice III

On lance une pièce de monnaie 10 fois de suite. Réaliser un programme qui estime les probabilités suivantes.

- Obtenir 3 faces exactement ?
- Obtenir 3 faces au plus ?
- Obtenir 3 faces au moins ?

Comparer avec les résultats trouvés l'année dernière en MF 100.

[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)

Exercice IV

Considérons l'ensemble $[\![1,4]\!] = \{1,2,3,4\}$ et l'ensemble de Σ_4 de ses permutations.

1. Définir un type permutations de $[\![1,4]\!] = \{1,2,3,4\}$.

2. Faire une fonction qui demande à l'utilisateur l'image de chaque élément de $[\![1,4]\!]$ et crée la permutation correspondante. La fonction vérifiera le caractère bijectif de ce qui est entré par l'utilisateur.

3. Faire une fonction d'affichage de la permutation.

4. Créer une fonction de composition des éléments de Σ_4.

5. Faire un programme utilisant ces fonctions.

Exercice V

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 une fonction `creerQ` qui prend comme argument deux entiers et retourne le nombre rationnel correspondant.

3. Faire une fonction `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).

4. Faire une fonction `approximerQ` qui prend comme argument un nombre rationnel et retourne un réel double précision l'approximant.

5. Faire une fonction `egalQ` qui prend comme arguments deux nombres rationnels et retourne 1 s'ils sont égaux et 0 sinon.

6. Faire une fonction `produitQ` qui prend comme arguments deux nombres rationnels et retourne leur produit.

7. Faire une fonction `divisionQ` qui prend comme arguments deux nombres rationnels, le deuxième étant non nul, et retourne leur quotient.

8. Faire une fonction `sommeQ` qui prend comme arguments deux nombres rationnels et retourne leur somme.

9. Faire une fonction `irreductibleQ` qui prend comme arguments un nombre rationnels et retourne un nombre rationnel égal sous forme irréductible.

10. Faire un programme qui simplifie l'expression suivante

$$\frac{\frac{1}{2}+\frac{1}{4}}{\frac{2}{9}+\frac{1}{7}}$$

affiche la fraction irréductible correspondante et une valeur approchée.