MPSI 2017-2018

Informatique

TP 2

Exercice I - Construction d'un fonction racine carrée

Partie mathématique. Soit a dans $\mathbb{R}^+$. Considérons la suite :

$$\begin{cases} u_{n+1} &= \frac{1}{2}\left(u_n + \frac{a}{u_n}\right) \\ u_0 &= a+1 \end{cases}$$

1. Montrer que la fonction de $\mathbb{R}^*$ dans $\mathbb{R}$ définie par $f(x) = \frac{1}{2}\left(x + \frac{a}{x}\right)$ est croissante sur $[\sqrt{a}; +\infty[$.
2. Montrer par récurrence que pour tout n de $\mathbb{N}$, u_n est bien définie et $u_n \geq \sqrt{a}$.
3. Montrer que (u_n) est décroissante. En déduire que (u_n) est convergente.
4. Montrer que la limite de (u_n) est $\sqrt{a}$.
5. Montrer que $\sqrt{a} \geq min(a, 1)$. En déduire que $|u_n - \sqrt{a}| \leq \frac{u_n^2 - a}{2.min(a,1)}$.

Partie informatique.

1. On considère que u_{1000} est une bonne approximation de $\sqrt{a}$. Écrire une fonction nommée *Racine* qui prend en entrée un réel positif et renvoie la valeur approchée de $\sqrt{a}$. On comparera les résultats donnés par votre fonction *Racine* et la fonction *sqrt* de Python.
2. Même exercice mais on s'arrête lorsqu'on atteint une précision inférieure à 10^{-12}

Exercice II - Nombres premiers

Partie mathématique. Soient n et d dans $\mathbb{N}$, on dit que d est un diviseur de n si et seulement si :

$$\exists k \in \mathbb{N},\ d.k = n$$

De plus n est un nombre premier si et seulement s'il admet exactement deux diviseurs.

1. Déterminer les diviseurs de 2, 8, 12, 1, 0. Est-ce que ce sont des nombres premiers ?
2. On suppose que $n \geq 2$, montrer que n est premier si et seulement s'il n'admet pas de diviseur entre 2 et $\sqrt{n}$.

Partie informatique.

1. Écrire une fonction booléenne (c'est-à-dire renvoyant true ou false) qui teste si le paramètre est un nombre premier ou non.
2. Écrire une fonction qui à partir d'une liste d'entiers renvoie la sous-liste des entiers qui sont premiers.
3. Écrire une fonction qui à partir d'un entier naturel non nul, le décompose en facteurs premiers et renvoie la liste de ces facteurs. Par exemple, pour 12, on doit renvoyer la liste $[2, 2, 3]$.

Mini-projet 1 : le mastermind

Le but est de créer un programme avec lequel on peut jouer au Mastermind. L'ordinateur choisit quatre couleurs différentes choisies dans la variable globale :

$$\text{Couleurs} = ['R','B','V','N','O','J']$$

et l'utilisateur doit les trouver en un minimum d'essai. A chaque essai, l'ordinateur donnera le nombre de pions de la bonne couleur et bien placés (BP) et le nombre de pions de la bonne couleur mais mal placés (MP). Exemple :

Pions cachés	R	V	B	N	
Essai 1	V	J	O	N	1BP, 1MP
Essai 2	R	B	V	N	2BP, 2MP
Essai 3	B	R	V	N	1BP, 3MP
Essai 4	R	V	B	N	4BP, 0MP

1. Écrire une fonction *CouleurDifférente* qui recoit une liste et qui vérifie que cette liste contient des éléments différents.
2. Écrire une fonction *Initialise* qui renvoie une liste contenant 4 couleurs différentes prises dans *Couleurs*. On vérifiera bien que ce choix est possible.

   Note : On pourra se servir de la fonction *randint(a, b)* de la biliothèque *random* qui renvoie un entier aléatoire entre a et b compris.
3. Écrire une fonction *Compare* qui à partir de deux listes indique le nombre de couleurs bien placées et de couleurs mal placées.
4. Écrire une fonction *DemandeCouleur* qui demande à l'utilisateur un choix de *NbDeCouleurs* couleurs et qui vérifie que les couleurs sont dans *Couleurs* et qu'elles sont bien différentes.
5. Écrire un programme permettant de jouer au Mastermind avec l'ordinateur.
6. Adapter le travail précédent pour autoriser à cacher plusieurs fois la même couleur. L'éditeur du jeu annonce "2401 combinaisons possibles". Expliquer le calcul menant à ce résultat.