TP Informatique 10 - Fonctions (suite)

Exercice 10.1

1. Réécrire sans regarder les précédents TP le programme qui demande à l'utilisateur de rentrer une valeur $x \in \mathbb{R}^*$ et un entier $n \in \mathbb{N}$, et qui rend x^n.
2. Modifier ce que vous venez d'écrire pour obtenir un programme disposant de la **fonction puissance** (à deux paramètres), et qui fait la même chose que le précédent.
3. Utiliser cette fonction pour écrire un programme qui demande à l'utilisateur de rentrer une valeur $x \in \mathbb{R}^*$ et un entier $n \in \mathbb{N}$ et qui calcule $\displaystyle\sum_{k=0}^{n} x^k$.
4. Ecrire un programme faisant la même chose mais sans utiliser de `FUNCTION`. Lequel est préférable pour l'ordinateur ?

Exercice 10.2

1. Ecrire une fonction `tirage` qui donne un numéro au hasard entre 1 et n, où n est quelconque.
2. On considère l'expérience suivante : on a une urne avec 100 boules numérotées de 1 à 100. On effecture 10 tirages avec remise et on s'intéresse au nombre X de boules paires obtenues.
   (a) Ecrire un programme qui simule cette expérience aléatoire. Il faudra afficher le résultat de chacun des 10 tirages. On utilisera bien sûr la fonction `tirage`
   (b) Ecrire un programme qui simule cette expérience aléatoire et rend le nombre X obtenu.
3. On considère l'expérience suivante : on a une urne U avec 100 boules numérotées de 1 à 100. On effectue un tirage avec remise, on note le numéro k de la boule obtenue ; on effectue alors un tirage secondaire dans une urne contenant k boules numérotées de 1 à k. On recommence : on tire dans l'urne U, on note le numéro obtenu, on tire alors dans une urne ne contenant que les boules de 1 jusqu'à ce numéro... On recommence 10 fois.
   (a) Ecrire un programme qui simule cette expérience aléatoire : il faudra afficher, pour les 10 étapes, le résultat du premier tirage et du tirage "secondaire".
   (b) On s'intéresse au nombre X de boules paires obtenues lors des tirages dits "secondaires". Ecrire un programme qui simule cette expérience aléatoire et rend le nombre X obtenu.

Exercice 10.3

La **fonction partie entière** $x \mapsto E(x)$ est prédéfinie en Turbo Pascal, c'est `trunc` (car $E(x)$ c'est le réel x qui est "tronqué"). Dans cet exercice, on n'utilisera PAS la fonction `trunc`.

1. Ecrire une fonction `partie` qui a pour paramètre un réel $x \in \mathbb{R}$ et qui rend $E(x)$.
   Pour cela : commencer par le cas $x \geq 0$: pour trouver la partie entière de x, on regarde si 0 convient, c'est-à-dire $x < 1$. Si ce n'est pas vérifié, on regarde si $x < 2$, auquel cas, on aura $E(x) = 1$. Si ce n'est pas vérifié, on regarde si $x < 2$, ... et TANT QUE $x < k$, on recommence ! Pour le cas $x < 0$, un raisonnement similaire permettra de conclure.
2. En déduire une fonction qui a pour paramètre $x \in \mathbb{R}$ et qui rend $f(x) = E(x) + (x - E(x))^2$.
3. Ecrire un programme demandant un réel $a \in \mathbb{R}$ et un entier $n \in \mathbb{N}$ et affichant le n-ième terme de la suite $\begin{cases} u_0 = a \\ \forall n \in \mathbb{N},\ u_{n+1} = E(u_n) + (u_n - E(u_n))^2 \end{cases}$
4. Ecrire un programme demandant deux réels a et b, et un entier $n \in \mathbb{N}$, et affichant le n-ième terme de la suite $\begin{cases} u_0 = a \\ u_1 = b \\ \forall n \in \mathbb{N},\ u_{n+2} = u_{n+1} + E(u_n) + (u_n - E(u_n))^2 \end{cases}$