UNIVERSITÉ D'ORLÉANS **DEUG MIAS**
IF 3.01 - Informatique **Année 2001-2002**

Feuille d'exercices n° 7

Exercice 1

Écrire une fonction qui, pour un réel x donné en paramètre, retourne la valeur $\log(\sin(x))$ lorque $\sin(x) > 0$, et affiche le message « argument invalide » sinon.

Exercice 2

Écrire une fonction qui prend en argument deux entiers x et y, et retourne le résultat de la division entière de x par y lorsque y n'est pas nul, et 0 si y est nul.

Exercice 3

Définir la fonction `map_succeed` qui, étant données une fonction f et une liste l, retourne la liste des applications réussies de f sur l (autrement dit, si l'application de f sur un élément de la liste échoue, l'application de `map_succeed` continue quand même).

Exercice 4

On veut utiliser les entrées-sorties et les exceptions.

1. Écrire une fonction `nieme` qui, étant donnés un entier n et une liste d'entiers l, renvoie le n-ième élément de la liste. Tenir compte des listes trop courtes et des cas où $n < 1$, et lever des exceptions si nécessaire.
2. On veut maintenant *tracer l'évolution*[1] de cette fonction à l'aide d'affichages. Modifier la fonction `nieme` de façon à suivre :
   – le début de l'évaluation d'une fonction ;
   – les levées d'exceptions ;
   – le renvoi du résultat ;
   – les appels de fonctions ;
   – les retours d'appels de fonctions.
3. On veut maintenant *couper l'arbre des appels* à l'aide d'exceptions. Pour cela, on définit une nouvelle exception `Resultat` accompagnée d'un entier. On modifie ensuite la fonction `nieme` de telle sorte qu'elle lève une exception `Resultat` accompagnée d'un entier r, au lie de renvoyer r. Il faut ensuite rattraper l'exception et renvoyer le résultat. Modifier les affichages en conséquence.
4. Écrire une fonction `longueur` qui calcule la longueur d'une liste en coupant l'arbre des appels à l'aide d'une exception `Resultat`. Il est inutile de mettre des affichages.

---

[1] C'est à dire suivre les différentes fonctions appelées