TP n° 5

1 Listes

Exercice 1 : Opérateurs seq et $

Question 1.1 Générer une séquence S1 contenant les 10 premiers éléments de la suite u définie par : $u_0 = 1$ et $u_{i+1} = u_i + 7$.

Question 1.2 Générer à partir de S1 la séquence S2 contenant les 10 premiers éléments de la suite v définie par $v_i = (-1)^i * u_i$.

Question 1.3 Générer la séquence S3 contenant les éléments de rang pair de S2.

Question 1.4 Générer la séquence S4 contenant les éléments de S3 en sens inverse.

Question 1.5 Générer la séquence S5 contenant les évaluations décimales à 1,2,..50 chiffres de $\frac{4}{3}$.

Question 1.6 Générer la séquence S6 : -1, 2, -3, 4, -5, 6, -7, 8, -9, 10, -11, 12

Exercice 2 : Listes de listes

Question 2.1 Prévoir avant de les taper les réponses des instructions Maple suivantes :

```
> LL1 :=[ [seq(i,i=1..4)], [a,b,c,d], [seq( (2*i-1)/2,i=1..4)], [seq(f,i=1..4)]] ;
> LL1[2..4] ;
> nops(LL1) ;
> nops(LL1[1]) ;
> member(b,LL1,'pos') ;pos ;
> member(b,LL1[2],'pos2') ;pos2 ;
> LL1[2] := subsop(2=3,LL1[2]) ;
> LL1 ;
> LL1[1][2] := 7 ; LL1 ;
```

Question 2.2 Générer la liste LL1 dont les éléments de rang n sont des listes de la forme $[n, n+3, n+6]$, pour n de 1 à 10.

Question 2.3 Générer une liste de liste LL2 dont les éléments de rang n, m sont de la forme $x^n + m$, pour n, m de 1 à 5. Vérifier que LL2[2][3] = $x^2 + 3$. (Penser à l'ordre de génération).

2 Boucles

Exercice 3

Déterminer l'entier n à partir duquel la somme des n premiers entiers est supérieure à 1000.

Exercice 4

Question 4.1 Écrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne.

Question 4.2 Écrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : " Plus petit ! ", et inversement, " Plus grand ! " si le nombre est inférieur à 10.

Exercice 5

Les fonctions *irem* et *iquo* calculent le reste et le quotient (respectivement) de la division entière d'un nombre a par un nombre b. Ainsi `irem(22,5)` retourne `2` et `iquo(22,5)` retourne 4.

Question 5.1 Ecrire un algorithme permettant de compter le nombre de diviseurs d'un entier n.

Question 5.2 Réécrire l'algorithme de façon à obtenir le nombre de diviseurs pairs et le nombre de diviseurs impairs.

Exercice 6 : Distributeur de boisson

On désire développer un programme qui permet de simuler un distributeur de boissons dont le fonctionnement est décrit ci-dessous.
Le distributeur propose des boissons chaudes et froides. Les boissons chaudes proposées sont : café, chocolat et thé. Chacune d'elles pourra être sucrée ou non. Les boissons froides sont : coca ou jus d'orange. Le choix d'une boisson est effectué en sélectionnant le ou les boutons adéquats. Les boissons gazeuses valent 1 euro, les non gazeuses (froides) 0,5 euro. Les boissons chaudes valent 1 euro. Le supplément pour le sucre est de 0,2 euro. Une fois le choix d'une boisson fait, le prix s'affiche. Le paiement se fait par insertion de pièces de monnaie de 2 euros, 1 euro, 0,5 euro, 0,2 euro et 0,1 euro. L'insertion d'une pièce décrémente la somme restant à payer. Une fois que l'on a introduit suffisamment de pièces (il n'est pas possible de faire autrement), l'appareil retourne la monnaie de façon intelligente (en cherchant à ne pas rendre trop de petites pièces). Par exemple si la somme de 1 euro 40 doit être rendue, elle le sera par une pièce de 1 euro et deux pièces de 0,2 euro plutôt que par deux pièces de 0,5 euros et quatre pièces de 0,1 euro.

La simulation demandera à l'utilisateur sa boisson (avec le sucre éventuel), affichera le prix à payer, demandera les pièces insérées par l'utilisateur, puis affichera les pièces de monnaie rendues.

3 Listes et boucles

Exercice 7

Etant données deux listes $l1$ et $l2$, écrire les programmes permettant :
- d'écrire un par ligne et dans l'ordre de la liste les éléments de $l1$
- d'écrire un par ligne et dans l'ordre inverse de la liste les éléments de $l1$
- de compter le nombre d'occurrence d'un élément dans une liste
- de construire $l3$ comme l'union sans doublon de $l1$ et $l2$
- de construire $l3$ comme l'intersection de $l1$ et $l2$
- de vérifier que les éléments de $l1$ sont une et une seule fois dans $l2$
- de retirer un élément de toutes ses positions dans la liste $l1$

Exercice 8

Etant donnée une liste d'entiers L, écrire un programme qui permette de déterminer le nombre le plus grand dans L.

Exercice 9

Ecrire un programme qui teste si une chaîne de caractères est un palindrome (chaîne qui se lit

de la même manière dans les deux sens, par exemple "kayak").

Exercice 10

Ecrire un programme qui simule le jeu du pendu simplifié : Maple génère un nombre aléatoire, on a dix essais pour trouver ce nombre. A chaque proposition, le programme doit indiquer si l'on a gagné et, sinon, si le nombre à trouver est plus petit ou plus grand que le nombre proposé, ainsi que le nombre d'essais qu'il nous reste. On utilisera la fonction rand : Taper b :=rand(1..1000) : (b est alors un générateur de nombres aléatoires entre 1 et 1000). Pour générer un nombre aléatoire, il suffit alors de taper : b() :

Exercice 11

Soit le tableau de notes suivant :

	Prénom	Math	...	Info	Français
Dupond	Jean	12	...	15	10
...	...	...	...	...	...
Durand	Paul	5	...	9	...

Question 11.1 Donner la structure de listes permettant de représenter ce tableau. Les étudiants ne suivant pas tous les mêmes options, le tableau peut comporter des " trous ". Par exemple, l'élève Durand n'a pas de notes en français et cette matière ne doit donc pas apparaître pour Durand.

Question 11.2 Ecrire un programme qui, étant donnée un étudiant, détermine la moyenne de ses notes.

Question 11.3 Ecrire un programme qui permet de retrouver pour un étudiant sa note la plus basse et qui affiche le nom de la matière concernée.

Question 11.4 Ecrire un programme qui permet pour une matière de calculer la moyenne des notes des étudiants concernés.