

8

TP SUR LES LISTES

Exercice 1 :

Ecrire une fonction *prodScalaire(U,V)* qui prend en paramètres les listes des trois coordonnées des vecteurs de l'espace \vec{u} et \vec{v} et renvoie le produit scalaire de \vec{u} et \vec{v} .
Par exemple, *prodScalaire([2,-5,4],[-1,5,6])* renverra -3.

Indication :

Si $\vec{u}(x;y;z)$ et $\vec{v}(x';y';z')$, alors $\vec{u} \cdot \vec{v} = xx' + yy' + zz'$.

Exercice 2 :

1. Ecrire une fonction *aleat(n)* qui prend en paramètre un entier n strictement positif et qui renvoie une liste de n nombres entiers compris entre 0 et 100. Tester cette fonction.

Indications :

Initialiser la nouvelle liste comme une liste vide.

Importer le module *random* et utiliser la fonction *randint(0,100)* qui renvoie aléatoirement un entier compris entre 0 et 100.

2. Ecrire une fonction *affichage(L)* qui prend en paramètre une liste d'entiers L et affiche les termes de la liste L les uns en dessous des autres. La tester.
3. Ecrire une fonction *supDix(L)* qui prend en paramètre une liste d'entiers L et qui renvoie le nombre de valeurs supérieures ou égales à 10. La tester.

Exercice 3 :

1. Créer les listes : *couleur=["pique","coeur","carreau","trèfle"]* et *valeur=["1","2","3","4","5","6","7","8","9","10","valet","dame","roi"]*.
2. Ecrire une fonction *creatJeu()* ne prenant aucun paramètre, qui à partir de ces deux listes, renvoie une troisième liste représentant un jeu de 52 cartes :
["1 pique", "2 pique", "3 pique", ..., "1 cœur", ..., "1 carreau", ..., "1 trèfle", ..., "roi trèfle"]

Indication :

- Initialiser la nouvelle liste comme une liste vide
 - Utiliser deux boucles imbriquées : l'une parcourant la liste *couleur*, l'autre parcourant la liste *valeur*
 - à chaque tour de boucle, utiliser l'opérateur de concaténation $+$ afin de créer une chaîne de caractères constituée d'un élément de couleur et d'un élément de valeur qui sera rajoutée à la nouvelle liste (rappel : "1"+"pique" donne "1pique")
3. Dans la partie principale du programme, affecter à une liste *jeu* le résultat renvoyé par la fonction *creatJeu()*.
 4. Ecrire une fonction *echangeCartes(i,j)* qui prend en paramètres deux indices i et j et qui modifie la liste *jeu* en échangeant les cartes d'indices i et j .

Indication :

Pour échanger les 2 cartes, il ne suffit pas d'écrire *jeu[i]=jeu[j]*, puis *jeu[j]=jeu[i]* car lorsqu'on exécute l'instruction *jeu[i]=jeu[j]*, l'ancienne valeur de *jeu[i]* s'efface. Il faut donc d'abord stocker la valeur contenue dans *jeu[i]* dans une variable temporaire (souvent nommée *tmp*), puis remplacer *jeu[i]* par le contenu de *jeu[j]* et enfin mettre dans *jeu[j]* le contenu de *tmp*.

5. Ecrire une fonction *melangeJeu(n)* qui prend en paramètre un entier n et procède à n échanges aléatoires dans la liste *jeu*.

Indications :

Importer le module *random* et pour chaque échange, utiliser 2 fois la fonction *randint* afin d'obtenir 2 indices correspondants aux cartes à échanger. Puis appeler la fonction précédente.

Exercice 4 :

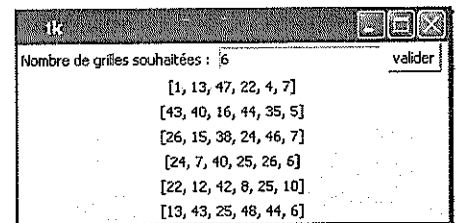
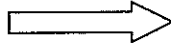
Une grille de loto simple est composée de 5 numéros, tous différents, compris entre 1 et 49 et d'un numéro « chance », compris entre 1 et 10. On souhaite programmer une application permettant de générer automatiquement n grilles de loto. Dans les deux premières questions, on va créer différentes fonctions permettant d'obtenir ces grilles, puis dans la question 3, on rajoutera une interface graphique.

1. Ecrire la fonction `grille()` qui renvoie une liste qui représentera une grille de loto.

Indications :

Utiliser la fonction `randint` pour générer aléatoirement chaque numéro ainsi que le numéro chance. Mais attention, ne pas oublier que les 5 premiers numéros doivent tous être différents, donc à chaque fois qu'on générera un nouveau numéro, on l'ajoutera à la grille uniquement s'il n'y est pas déjà (pour tester la non appartenance, on pourra utiliser l'instruction `not in`).

2. Ecrire la fonction `plusieursGrilles(n)` qui prend en paramètre un entier naturel n , génère n grilles de loto et affiche les différentes grilles générées.
3. Créer une interface graphique du type :



Indications :

- Créer l'interface de base avec un label, une zone de saisie et un bouton valider qui fera appel à la fonction `plusieursGrilles`.
- Modifier la fonction `plusieursGrilles` :
 - n ne doit plus être en paramètre, mais doit contenir la valeur saisie (voir TP6 ex 3 pour la récupération de la valeur saisie dans la zone de saisie, mais attention il faudra la convertir en entier)
 - à chaque tour de boucle, on crée un nouveau label qui contient une grille de loto générée par la fonction `grille()` et qui vient se placer en-dessous des précédents.