Structures de contrôle et raffinages

Objectifs

— Raffiner des problèmes simples
— Écrire quelques algorithmes simples
— Manipuler les structures de controle

1 Structures de contrôle

Exercice 1 : Quelques statistiques
L'objectif de cet exercice est de calculer quelques statistiques sur une série de valeurs réelles. Cette série est lue au clavier. On considère qu'elle se termine par la valeur nulle (0) qui ne fait pas partie de la série.

1.1 Écrire un programme qui détermine la moyenne des valeurs de la série.

1.2 En plus de la moyenne, on veut connaître la plus petite et la plus grande valeur de la série.

2 Liste et Raffinages

Exercice 2 : Tri par sélection
Soit $A[1..N]$ un vecteur de N entiers relatifs quelconques, l'objectif est de trier le vecteur A en utilisant le tri par sélection. Le vecteur A est trié si $A[1] \leq A[2] \leq \cdots \leq A[N]$.

Le tri par sélection est un tri en $(N-1)$ étapes. L'étape i consiste à ranger à sa place le i^{e} plus petit élément du vecteur.
Exemple : Voici les différentes valeurs du vecteur 8 2 9 5 1 7 après chaque étape (la partie encadrée correspond à la partie du vecteur déjà traitée et donc triée) :

vecteur initial	:	8 2 9 5 $\underline{1}$ 7
après l'étape 1	:	$\boxed{1}$ $\underline{2}$ 9 5 8 7
après l'étape 2	:	$\boxed{1\ 2}$ 9 $\underline{5}$ 8 7
après l'étape 3	:	$\boxed{1\ 2\ 5}$ 9 8 $\underline{7}$
après l'étape 4	:	$\boxed{1\ 2\ 5\ 7}$ $\underline{8}$ 9
après l'étape 5	:	$\boxed{1\ 2\ 5\ 7\ 8\ 9}$

2.1 Implanter l'algorithme du tri par sélection.

Exercice 3 : Tri par insertion séquentielle
Soit $A[1..N]$ un vecteur de N entiers relatifs quelconques, l'objectif est de trier le vecteur A. Le vecteur A est trié si $A[1] \leq A[2] \leq \cdots \leq A[N]$.

Le tri utilisé est le tri par insertion séquentielle. C'est un tri en $(N-1)$ étapes. L'étape i consiste à placer le $(i+1)^{\text{e}}$ élément du vecteur à sa place dans le sous-vecteur $A(1..i+1)$ sachant que $A(1..i)$ a été trié par les étapes précédentes. Dans le cas du tri par insertion séquentielle, la recherche de la position d'insertion, se fait séquentiellement en comparant successivement l'élément à insérer aux i premiers éléments du vecteur.
Exemple : Voici les différentes valeurs du vecteur 8 2 9 5 1 7 après chaque étape (la partie encadrée correspond à la partie du vecteur déjà traitée et donc triée) :

vecteur initial	:	[8] 2 9 5 1 7
après l'étape 1	:	[2 8] 9 5 1 7
après l'étape 2	:	[2 8 9] 5 1 7
après l'étape 3	:	[2 5 8 9] 1 7
après l'étape 4	:	[1 2 5 8 9] 7
après l'étape 5	:	[1 2 5 7 8 9]

3.1 Implanter l'algorithme du tri par insertion.

3.2 En conservant le principe du tri par insertion, expliquer comment améliorer l'efficacité de cet algorithme.

3 Raffinages

Exercice 4 : Jeu du devin
Le jeu du devin se joue à deux joueurs. Le premier joueur choisit un nombre compris entre 1 et 999. Le second doit le trouver en un minimum d'essais. À chaque proposition, le premier joueur indique si le nombre proposé est plus grand ou plus petit que le nombre à trouver. En fin de partie, le nombre d'essais est donné.
Indication : On suppose qu'il existe une fonction qui permet d'obtenir un nombre aléatoire compris entre 1 et 999. En Python, cette fonction s'appelle `randint` (pour « *random integer* » = nombre entier aléatoire) et s'utilise ainsi :

```
nombre = randint(1, 999)        # nombre référence le nombre aléatoire
```

La fonction `randint` est définie dans le module `random`. Pour y avoir accès, il faut ajouter (en début de fichier) :

```
from random import randint
```

4.1 *La machine fait deviner le nombre.* Écrire un programme dans lequel la machine choisit un nombre et le fait deviner à l'utilisateur.

4.2 *La machine joue.* Écrire un programme dans lequel l'utilisateur choisit un nombre et la machine doit le trouver. Pour chaque nombre proposé, l'utilisateur indique s'il est trop petit (`'p'` ou `'P'`), trop grand (`'g'` ou `'G'`) ou trouvé (`'t'`, `'T'`).

Indication : On utilisera une recherche par dichotomie pour trouver le nombre.

4.3 *Le programme complet.* Écrire le programme de jeu qui donne le choix à l'utilisateur entre deviner ou faire deviner le nombre.

4.4 *On peut recommencer.* Compléter le programme précédent pour que l'ordinateur propose de faire une nouvelle partie lorsque la précédente est terminée.