



Travaux dirigés – feuille 1

Exercice n°1

Écrire un programme C se comportant comme une calculatrice, c'est-à-dire bouclant sur les actions :

- lecture de deux entiers ;
- lecture d'un opérateur binaire parmi +, -, *, / et % (attention aux spécificités de chaque opérateur) ;
- calcul de la valeur de l'expression ;
- affichage du résultat à l'écran ;
- demande à l'utilisateur s'il veut ou pas recommencer (l'utilisateur saisit 1 s'il veut recommencer, 0 sinon).

Exemple :

Donner un entier : 2	Donner un entier : 2
Donner un entier : 3	Donner un entier : 3
Donner un opérateur : +	Donner un opérateur : *
La somme de 2 et 3 est égale à 5	La multiplication de 2 et 3 est égale à 6
Autre opération ? (0/1)1	Autre opération ? (0/1)0

Exercice n°2

- a) Écrire un programme permettant à l'ordinateur de trouver quel nombre entre (1 et 100) a choisi l'utilisateur. Pour trouver le nombre, la machine fera des propositions successives à l'utilisateur. À chacune de ces propositions, l'utilisateur devra répondre en indiquant si le nombre à rechercher est inférieur, égal ou supérieur au nombre proposé (voir exemple). Quand le programme a trouvé, il doit afficher le nombre de coups utilisés pour trouver le nombre.

Exemple de dialogue affiché sur l'écran au cours d'une exécution du programme (ce que tape l'utilisateur est en gras)
L'utilisateur a choisi ici le nombre 59 :

```
Pensez à un entier entre 1 et 100.
Tapez -1 si votre nombre est inférieur à 50, 1 s'il est supérieur, 0 s'il est égal
: 1
Tapez -1 si votre nombre est inférieur à 75, 1 s'il est supérieur, 0 s'il est égal
: -1
Tapez -1 si votre nombre est inférieur à 62, 1 s'il est supérieur, 0 s'il est égal
: -1
Tapez -1 si votre nombre est inférieur à 56, 1 s'il est supérieur, 0 s'il est égal
: 1
Tapez -1 si votre nombre est inférieur à 59, 1 s'il est supérieur, 0 s'il est égal
: 0
J'ai trouvé en 5 coups !
```

- b) Apporter les modifications nécessaires à votre code afin de limiter le nombre d'essais de l'ordinateur pour trouver le nombre.

Exercice n°3

- a) Écrire un programme qui permet de saisir un nombre entier puis de déterminer et d'afficher la somme de ses chiffres.
- b) Écrire un programme qui saisit un nombre entier positif puis détermine s'il s'agit d'un nombre idiot. Un nombre (exprimé en base 10) est dit "idiot", s'il est égal à la somme des cubes de chacun de ses chiffres.
Exemple : 153 est un nombre idiot car $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$
- c) Modifier le code pour que le programme affiche tous les nombres idiots compris entre 1 et 1000.

Exercice n°4 : recherche séquentielle

- a) Écrire un programme qui recherche si une valeur entière donnée est contenue dans un tableau d'entiers (de 100 cases maximum) non trié et supposé comporter nbElem cases remplies.
- b) Écrire un programme qui recherche si une valeur entière donnée est contenue dans un tableau d'entiers (de 100 cases maximum) trié (selon l'ordre croissant des valeurs qu'il contient) et supposé comporter nbElem cases remplies.

Exercice n°5 : recherche dichotomique

Écrire un programme qui recherche, par dichotomie, si une valeur entière donnée est contenue dans un tableau d'entiers (de 100 cases maximum) trié (selon l'ordre croissant des valeurs qu'il contient) et supposé comporter nbElem cases remplies.

Principe de la méthode de la "recherche dichotomique".

Cette méthode ne s'applique que sur une liste d'éléments triés.

Le principe de la recherche dichotomique consiste à comparer x avec l'élément situé au milieu m de la liste t :

- si $x == t[m]$, on a trouvé une solution et la recherche s'arrête ;
- si $x < t[m]$, x ne peut se trouver que dans la moitié gauche de la liste t puisque celle-ci est triée par ordre croissant ; on recommence alors la recherche de la même manière uniquement dans la moitié gauche de la liste ;
- si $x > t[m]$, x ne peut se trouver que dans la moitié droite de la liste t ; on recommence alors la recherche uniquement dans la moitié droite de la liste.

A chaque fois, on poursuit donc la recherche en diminuant de moitié le nombre d'éléments restant à traiter.

Exercice n°6

- a) Écrire un programme permettant de :
- saisir une suite de caractères donnée par l'utilisateur pouvant contenir des lettres minuscules et majuscules ;
 - modifier la chaîne attentionnée, en transformant chaque caractère majuscule en minuscule ; pour convertir un caractère majuscule vers le caractère minuscule correspondant, il suffit de faire un décalage de son code ASCII. Le caractère 'A' a pour code 65 (B a pour code 66, ..., Z a pour code 90) et le caractère 'a' a pour code 97 ('b' a pour code 98, ..., 'z' a pour code 122) ;
 - afficher la chaîne saisie ainsi que le nombre de caractères la composant.

Exemple de trace :

Saisir votre chaîne : **bonjour Hector, comment allez VOUS ?**

Vous avez saisi : bonjour hector, comment allez vous ?, composée de 36 caractères

- b) Compléter le programme pour qu'il détermine si une chaîne, supposée ne pas contenir d'espace, est, ou pas, un palindrome. Un palindrome est un mot qui se lit de manière identique de la gauche vers la droite et inversement.
Exemples : les chaînes "ada", "abccba", "laval", "ressasser" sont des palindromes
- c) Modifier le programme pour qu'il détermine si une chaîne, pouvant contenir des espaces, est, ou pas, un palindrome.
Exemples : La chaîne "ab c c ba" est un palindrome. En revanche, la chaîne "ab ccba" ne l'est pas