

EXERCICE 29

Ecrire une fonction qui permet de calculer l'occurrence d'un entier A dans un tableau d'entiers T de taille N.

EXERCICE 30

Ecrire une procédure qui permet de réaliser une décomposition élémentaire d'un entier A en ses diviseurs premiers.

Exemple :

24 est décomposé comme suit : $24 = 2 \times 2 \times 2 \times 3$, la procédure affiche : 1, 2, 2, 2, 3

42 est décomposé comme suit : $42 = 2 \times 3 \times 7$, la procédure affiche : 1, 2, 3, 7

17 est décomposé comme suit : $17 = 1 \times 17$, la procédure affiche : 1, 17

EXERCICE 31

Ecrire une procédure qui permet de permuter deux entiers A et B. Utiliser cette procédure dans un programme.

EXERCICE 32 (facultatif)

Ecrire une procédure qui transforme un entier A d'une manière symétrique.

C'est-à-dire, si A=3524 alors A devient = 4253. Utiliser cette procédure dans un programme.

EXERCICE 33 (facultatif)

Ecrire une procédure qui permet d'inverser une chaîne de caractère ch.

EXERCICE 34 (facultatif : jeu vache-taureau)

Coder le fameux jeu vache-taureau dont l'objectif est de trouver le nombre (à quatre chiffres) générés aléatoirement par l'ordinateur. Ce nombre doit contenir des chiffres différents.

L'utilisateur a 5 ou 6 essais pour trouver le nombre. A chaque essai, le programme renvoie autant de « taureaux » pour des chiffres qui existent et à la bonne position, et autant de « vaches » pour des chiffres qui existent à des mauvaises positions.

Exemple : Nombre recherché égal à 4157 : pour l'essai 1234 le programme retourne 2 vaches et pour l'essai 4567, le programme retourne 1 vache et 2 taureaux.

Indications :

```
int random(int amax){
    int a;
    a=((int) (rand()%1000)*(amax-1)/1000);
    return(a);
}
```

La fonction précédente permet de demander à l'ordinateur de générer aléatoirement un entier entre 0 et amax. L'utilisation de la fonction rand() nécessite les bibliothèques <stdlib.h> et <time.h>.