Exercices: Série 4 - Tableaux - ICC-C 2025-2026

1. Échauffement : produit des éléments d'un tableau

Écrire le programme `produit.c` qui :

1. définit un tableau d'entiers contenant les nombres $5, 3, 65, 17, 2, 5, 3$;
2. au moyen d'une boucle, affiche chaque élément du tableau sur une ligne distincte ;
3. calcule, au moyen d'une boucle, le produit p de ces nombres, puis l'affiche ;
4. déclare un autre tableau, de `double`, de la même taille ;
5. remplit, au moyen d'une boucle, ce deuxième tableau avec les *logarithmes* des éléments du premier tableau ($\ln 5$, $\ln 3$, etc.) ;
6. calcule, au moyen d'une boucle, la *somme* s de ces nombres, puis affiche s ainsi que e^s avec 10 chiffres derrière la virgule.

Utilisez les fonctions `log` et `exp` de `math.h`.

Exemple d'exécution du programme :

```
5
3
65
17
2
5
3
p = 497250
s = 13.1168481967
e^s = 497249.9999999986
```

2. Produit scalaire

Soient les vecteurs d'entiers suivants :

$$\boldsymbol{a} = (5, 3, 65, 17, 2, 5, 3)$$
$$\boldsymbol{b} = (4, 61, 456, 12, 3, 1, 5)$$
$$\boldsymbol{c} = (8, 45, 3, 4)$$
$$\boldsymbol{d} = (89, 4, 74, 1)$$

Écrire le programme `scalaire.c` qui :

1. définit une fonction `scalaire`, qui prend deux tableaux d'entiers de même taille, ainsi que leur taille commune, en paramètres, et qui renvoie le produit scalaire de ces deux vecteurs ;
2. définit les 4 vecteurs ci-dessus, et affiche les produits scalaires $\boldsymbol{a} \cdot \boldsymbol{b}$ et $\boldsymbol{c} \cdot \boldsymbol{d}$.

Pour rappel, le produit scalaire est la somme des produits des éléments deux-à-deux :

$$\boldsymbol{v} \cdot \boldsymbol{u} = \sum_{i=1}^{n} v_i \cdot u_i$$

Exemple d'exécution du programme :

```
a . b = 30073
c . d = 1118
```

3. Cadenas

On a placé un cadenas avec n chiffres de 0 à 6 sur notre vélo, mais on a perdu le code ! Il va falloir essayer tous les codes possibles pour le retrouver. Afin de ne pas oublier de possibilités, on va écrire un petit programme `cadenas.c` qui énumère tous les codes possibles.

On représente un code possible par un tableau de n entiers. Chaque entier est compris entre 0 et 6 inclus.

Écrire une fonction `suivant` qui prend un code en paramètre, ainsi que le nombre de chiffres qui le composent, et qui calcule le code suivant à tester. Après 044, on veut tester 045, puis 046, puis 050, puis 051, etc. La fonction `suivant` doit *modifier* son argument pour passer au suivant. Elle doit renvoyer `true` si elle a pu passer à la combinaison suivante, et `false` s'il n'y a plus de combinaison à tester (dans le cas de 3 chiffres, si la combinaison reçue était 666). Elle procède de la façon suivante :

1. Si le dernier chiffre n'est pas 6, l'augmenter de 1 puis renvoyer `true`.
2. Sinon, remplacer le dernier chiffre par 0, puis :
3. Si l'avant-dernier chiffre n'est pas 6, l'augmenter de 1 puis renvoyer `true`.
4. Sinon, remplacer l'avant-dernier chiffre par 0, puis :
5. etc.
6. Si on a épuisé tous les chiffres, renvoyer `false`.

Écrire une fonction `show` qui prend un code en paramètre et qui l'affiche à l'écran, sur une ligne.

Écrire la fonction `main` qui :

1. Demande à l'utilisatrice ou l'utilisateur le nombre de chiffres du cadenas ;
2. Définit un tableau d'entier de cette taille, et l'initialise avec des 0 partout ;
3. Appelle successivement `show` et `suivant` jusqu'à ce que `suivant` renvoie `false`, afin d'afficher à l'écran toutes les combinaisons possibles.

Exemple d'exécution du programme :

```
Combien de chiffres : 2
00
01
02
03
04
05
06
10
11
12
...
56
60
61
62
63
64
65
66
```

Votre programme fonctionne fonctionne-t-il avec un cadenas à *0 chiffre* ? Qu'affiche-t-il ?