

Programmation : TP 3

Juliusz Chroboczek

3 octobre 2022

Premières boucles

Exercice 1. Écrivez un programme `vertical.c` qui demande à l'utilisateur un entier n et affiche une colonne de « * » de longueur n .

Exercice 2. Écrivez un programme `horizontal.c` qui demande à l'utilisateur un entier n et affiche une ligne de « * » de longueur n .

Exercice 3. Écrivez un programme `carres.c` qui affiche les carrés des 10 premiers nombres naturels, c'est-à-dire la suite d'entiers 1, 4, 9 ... 100.

Exercice 4. Écrivez un programme `multiples.c` qui demande à l'utilisateur un entier n puis affiche les 10 premiers multiples de n . Par exemple, si l'utilisateur entre 7, votre programme devra afficher la suite 7, 14, 21 ... 70.

Boucles avec accumulation

Exercice 5. Écrivez un programme `somme-cubes.c` qui demande à l'utilisateur un entier n puis qui affiche la somme des cubes des n premiers nombres entiers. Par exemple, si l'utilisateur entre 5, votre programme devra afficher 225, car

$$\begin{aligned}\sum_{k=1}^5 k^3 &= 1^3 + 2^3 + 3^3 + 4^3 + 5^3 \\ &= 1 + 8 + 27 + 64 + 125 \\ &= 225\end{aligned}$$

Exercice 6. Écrivez un programme `somme.c` qui demande à l'utilisateur un entier n , puis qui lit n entiers et affiche leur somme et leur moyenne. Vous n'êtes pas autorisés à accumuler dans une variable de type `double`. Testez notamment sur la suite (9, 10), le résultat doit être 9.5 et pas 9.

Boucles avec *flags*

Exercice 7. Écrivez un programme `sept.c` qui demande à l'utilisateur un entier n puis qui lit n entiers et indique à l'utilisateur si le nombre 7 se trouvait parmi ces n entiers.

Exercice 8. Écrivez un programme `premier.c` qui demande à l'utilisateur un entier n puis qui indique à l'utilisateur si cet entier est premier.

Boucles imbriquées.

Exercice 9. Écrivez un programme `carre1.c` qui demande à l'utilisateur un entier n et affiche un carré plein de « * » de côté n .

```
*****
*****
*****
*****
*****
```

Exercice 10. Écrivez un programme `carre2.c` qui demande à l'utilisateur un entier n et affiche un carré creux de « * » de côté n .

```
*****
*   *
*   *
*   *
*****
```

Exercice 11. Écrivez un programme `triangle1.c` qui demande à l'utilisateur un entier n et affiche un triangle plein de « * » de largeur n avec la pointe en haut à gauche.

```
*
**
***
****
*****
```

Exercice 12. Écrivez un programme `triangle2.c` qui demande à l'utilisateur un entier n et affiche un triangle plein de « * » de largeur n avec la pointe en bas à droite.

```
*****
****
***
**
*
```