

TP - Premières boucles `for`

Objectif(s)

- ★ Comprendre le fonctionnement des boucles `for`
- ★ savoir utiliser les boucles
- ★ savoir choisir quel type de boucle utiliser

Exercice 1 – Affichages de nombres

Question 1

Écrivez une fonction `affiche_nombres` qui prend en argument une valeur (`int`) n et affiche la valeur de n suivi de :, suivi tous les entiers de 1 à n , séparés par des espaces. Par exemple, sur l'entrée 5, la fonction doit afficher :

```
5: 1 2 3 4 5
```

Ecrivez une fonction `main` pour tester votre fonction.

Pour tester votre fonction `affiche_nombres`, vous l'appelerez dans le `main` avec différentes valeurs. Par exemple : 5, 8, 1, 0, -5. Pour chacun des cas, assurez-vous que vous comprenez bien le comportement de votre programme.

Question 2

Écrivez une fonction `affiche_nombres_pairs` qui prend en argument une valeur (`int`) n et affiche tous les entiers positifs pairs plus petits ou égaux à n , séparés par des espaces.

Pour tester votre fonction `affiche_nombres_pairs`, vous l'appelerez dans le `main` avec différentes valeurs. Par exemple : 6, 8, 9, -5. Pour chacun des cas, assurez-vous que vous comprenez bien le comportement de votre programme.

Question 3

Écrivez une fonction `affiche_nombres_decroissants` qui prend en argument une valeur (`int`) n et affiche tous les entiers positifs de n à 1, séparés par des espaces.

Pour tester votre fonction `affiche_nombres_pairs`, vous l'appelerez dans le `main` avec différentes valeurs. Par exemple : 6, 8, 9, -5. Pour chacun des cas, assurez-vous que vous comprenez bien le comportement de votre programme.

Exercice 2 – Compteurs

Question 1

Écrivez la fonction `factorielle` qui calcule et retourne la valeur de $n!$ pour un entier n .

Testez votre fonction dans une fonction dédiée avec les valeurs suivantes : 4, 0, 1, 2, 10. Pour rappel : $10! = 3628800$

Calculez les valeurs suivantes : $14!$, $15!$, $16!$, $17!$. Que pensez vous des résultats ?

Question 2

Écrivez une fonction `somme_impairs` qui calcule la somme des entiers impairs inférieurs à un paramètre n .

Calculer "à la main" la somme des nombres impairs de 1 à 5. Vérifiez que votre programme calcule la même valeur.

Proposez quelques valeurs complémentaires pour tester votre fonction. Vérifiez que votre fonction donne les bons résultats.

Renforcements

Remarque:

Il peut être utile d'avoir fait les renforcements du tp précédent pour faire ceux-ci. En fait, il peut même être pertinent de faire systématiquement les renforcements des TPs précédents avant de faire ceux du jour...

Exercice 3 – Compter

Question 1

Écrivez une fonction `compter_multiples` qui prend deux entiers *nb* et *mult* en argument, et retourne le nombre de multiples de *mult* entre 1 et *nb*. On le fera avec une boucle `for` même s'il est possible de prévoir le résultat mathématiquement...

Question 2

Comme d'habitude, testez votre fonction avec une fonction `test_compter_multiples`.

Question 3

Écrivez une fonction qui compte le nombre d'année bissextiles entre deux années données en paramètre. Il est pertinent d'appeler la fonction écrite lors d'un TD précédent...

Quand vous avez tout fini, vous pourriez vous entraîner avec le gameshell en consultant cette activité.