

Manipulation de boucles [lp02]

Exercices résolus

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 17 mai 2018

Table des matières

1 Compter de 1 à n / pgcompter	2
1.1 Compter de 1 à 10	2
1.2 Compter de 1 à n	3
1.3 Compter de 1 à n (ordre décroissant)	4
2 Références générales	4

C - Compter de 1 à n (Solution)



Utilise Structures répétitives ■
Durée estimée 15 min ■



Objectif

Cet exercice compte de 1 à n . Pour qu'il soit **profitable**, essayez de le faire par vous même avant de visualiser et/ou de télécharger les solutions.

1 Compter de 1 à n / pgcompter

1.1 Compter de 1 à 10

Imaginons que l'on veuille afficher tous les nombres de 1 à 10 :

```
1 2 3 4 5 6 7 8 9 10
```

((alg)) Algorithme (mauvaise solution)

Début

```
| Afficher ( 1 )  
| Afficher ( 2 )  
| Afficher ( 3 )  
| Afficher ( 4 )  
| Afficher ( 5 )  
| Afficher ( 6 )  
| Afficher ( 7 )  
| Afficher ( 8 )  
| Afficher ( 9 )  
| Afficher ( 10 )
```

Fin

Explication

C'est long à écrire, très peu souple et cela ne fonctionne que pour 10.



Une bonne solution : la boucle

Elle va nous permettre d'obtenir un algorithme qui s'adapte à la limite du décompte. Posons-nous les bonnes questions :

- Quelle est la tâche à répéter ?
Réponse : Afficher un nombre.
- Comment savoir si on continue ?
Réponse : On arrête quand « 10 » est affiché.
- Comment afficher à chaque fois un nombre différent ?
Réponse : Au travers d'une variable qui prendra toutes les valeurs de 1 à 10.
- Connaît-on le nombre d'exécution de la boucle ?
Réponse : Oui c'est 10. La variable de contrôle va évoluer de 1 à 10 ce qui tombe bien car c'est justement le nombre à afficher à chaque fois.



Écrivez une répétitive **Pour** de variable de boucle **j** (par exemple) qui varie de 1 à 10 et dans laquelle, affichez la valeur de **j**.



Validez votre code avec la solution.

Solution C @[pgcompter1.c]

```
int j;  
for (j=1; j<=10; ++j)  
{  
    printf("%d ",j);  
}  
printf("\n");
```

Solution commentée

On obtient une solution courte et lisible car l'en-tête du **Pour** indique clairement comment va évoluer la boucle (valeur de départ, valeur finale et pas).

1.2 Compter de 1 à n

Supposons maintenant que l'on veuille afficher les nombres de 1 à **n** où **n** est une valeur donnée par l'utilisateur. Ici rien de plus simple : il suffit de saisir cette valeur et de l'utiliser comme limite de boucle.



Modifiez votre programme de sorte qu'il saisisse un entier dans **n**.

Affichez l'invite :

Compter jusqu'à?



Modifiez votre répétitive **Pour** de sorte que **j** parcoure les entiers de 1 à **n**.



Testez. Exemple d'exécution :

Compter jusqu'à? 15

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15



Validez votre programme avec la solution.

Solution C @[pgcompterN.c]

```
#include <stdio.h>  
int main(void)  
{  
    int n;  
    printf("Compter jusqu'à? ");  
    scanf("%d",&n);  
    int j;  
    for (j=1; j<=n; ++j)  
    {  
        printf("%d ",j);  
    }  
    printf("\n");  
}
```



Que se passe-t-il si l'utilisateur entre une valeur négative ?

Solution simple

On n'entre pas dans la boucle : donc on n'affiche rien.

1.3 Compter de 1 à n (ordre décroissant)



Complétez maintenant votre programme pour qu'ensuite il compte de 1 à **n** en **ordre décroissant**.

Aide méthodologique

Copiez/collez la boucle **Pour** sur **j** puis modifiez-la pour compter de **n** à 1 par pas de -1 .



Testez.



Validez votre programme avec la solution.

Solution C

@[pgcompterX.c]

```
#include <stdio.h>
int main(void)
{
    int n;
    printf("Compter jusqu'à? ");
    scanf("%d",&n);
    int j;
    for (j=1; j<=n; ++j)
    {
        printf("%d ",j);
    }
    for (j=n; j>=1; --j)
    {
        printf("%d ",j);
    }
    printf("\n");
}
```

2 Références générales

Comprend [Chaty-PG1 :c3 :ex1] ■