

Afficher les entiers pairs [lp02]

Exercice résolu

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 17 mai 2018

Table des matières

1	Afficher les entiers pairs / pgaffPairs	2
1.1	Afficher n entiers pairs	2
1.2	Afficher les entiers pairs (1)	3
1.3	Afficher les entiers pairs (2)	3
2	Références générales	4

Java - Afficher les entiers pairs (Solution)



Utilise Structures répétitives ■
Durée estimée 20 min ■



Objectif

Cet exercice affiche les entiers pairs. Pour qu'il soit **profitable**, essayez de les faire par vous même avant de les visualiser et/ou de télécharger les solutions.

1 Afficher les entiers pairs / pgaffPairs

1.1 Afficher n entiers pairs

Affichons les `n` premiers nombres pairs :

```
n? 10
2 4 6 8 10 12 14 16 18 20
```

Pour ce, on peut partir de la solution de l'exercice @Compter de 1 à n] en modifiant la valeur à afficher (ici le double de la variable de contrôle) dans le corps de la boucle.



Écrivez un programme qui saisit un entier dans `n`.
Affichez l'invite :

```
n?
```



Écrivez une répétitive `Pour` de variable de boucle `j` (par exemple) qui varie de 1 à `n` (inclus) et dans laquelle, affichez la valeur de `2*j`. Rappel : la variable de contrôle compte le nombre d'itérations.



Testez.



Validez votre programme avec la solution.

Solution Java @[pgaffPairs1.java]

```
import java.util.Scanner;

class PGAffPairs1 {

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int n;
        System.out.print("n? ");
        n = input.nextInt();
        for (int j = 1; j <= n; ++j)
        {
            System.out.print((2 * j) + " ");
        }
        System.out.print("\n");
    }
}
```

1.2 Afficher les entiers pairs (1)

Cette fois-ci on souhaite afficher **uniquement** les nombres **pairs** compris entre 1 et n . Par exemple, pour $n = 10$, c'est : 2, 4, 6, 8, 10. Notez que n peut être impair. Si n vaut 11, l'affichage est le même que pour 10.



Peut-on utiliser un **Pour** ?

Solution simple

Oui. De 1 à n , il y a exactement « $n \text{ div } 2$ » nombres à afficher. La difficulté vient du lien à faire entre la variable de contrôle et le nombre à afficher. Du problème précédent, le double de celle-ci convient.



Modifiez alors votre boucle pour qu'il affiche les entiers pairs compris entre 1 et n .



Testez.



Validez votre programme avec la solution.

Solution Java

@[pgaffPairs2a.java]

```
import java.util.Scanner;

class PGAffPairs2a {

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int n;
        System.out.print("n? ");
        n = input.nextInt();
        int nsup = n / 2;
        for (int j = 1; j <= nsup; ++j)
        {
            System.out.print((2 * j) + " ");
        }
        System.out.print("\n");
    }
}
```

1.3 Afficher les entiers pairs (2)



Existe-t-il une solution plus simple ?

Solution simple

Oui. Elle consiste à garder le lien entre la variable de contrôle et le nombre à afficher. Dans ce cas, on commence à 2 et le pas doit être de 2.



Complétez votre programme pour qu'ensuite il affiche les entiers pairs compris entre 1 et `n` via un pas de 2 de la variable de boucle.



Testez.



Validez votre programme avec la solution.

Solution Java @[pgaffPairs2b.java]

```
import java.util.Scanner;

class PGAffPairs2b {

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int n;
        System.out.print("n? ");
        n = input.nextInt();
        for (int j = 2; j <= n; j += 2)
        {
            System.out.print(j + " ");
        }
        System.out.print("\n");
    }
}
```

2 Références générales

Comprend [Chaty-PG1 :c3 :ex1] ■