

# Distance entre deux points [bs10] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel  algoprog  Version 13 mai 2018

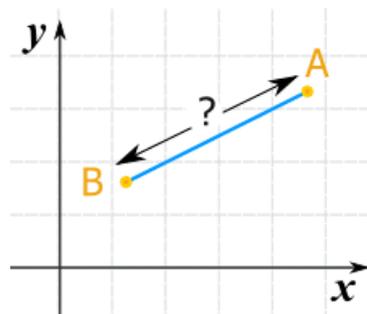
## Table des matières

<b>1</b>	<b>Distance entre deux points / pgdistance</b>	<b>2</b>
1.1	Calcul de la distance . . . . .	2
1.2	Calcul de la cote . . . . .	2
1.3	Validation . . . . .	3
<b>2</b>	<b>Références générales</b>	<b>4</b>

## Java - Distance entre deux points (Solution)

 Mots-Clés Structures de base ■  
Difficulté ●○○ (20 min) ■

 **Objectif**  
Cet exercice calcule la distance entre deux points du plan ainsi que la cote de chacun des points dans l'espace. (image : <https://www.mathsisfun.com>)



...(énoncé page suivante)...

# 1 Distance entre deux points / pgdistance

## 1.1 Calcul de la distance



Écrivez un programme qui saisit les coordonnées de deux points du plan  $(x_1, y_1)$  dans `x1`, `y1` pour le premier et  $(x_2, y_2)$  dans `x2`, `y2` pour le deuxième. Toutes les données sont réelles. Affichez les invites :

```
Premier point?
Deuxième point?
```



Calculez la distance entre les deux points définie par :

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

### Outil Java

Il n'existe pas de fonction carré (utilisez l'opérateur produit `*` ou la fonction `Math.pow(x, n)` de  $x^n$ . L'opération  $\sqrt{x}$  s'écrit `Math.sqrt(x)`.



Affichez (où `[x]` désigne le contenu de `x`) :

```
La distance est [...]
```



Testez. Exemple d'exécution :

```
Premier point? -4 3
Deuxieme point? 3.1 4.1
La distance est 7.18471
```

## 1.2 Calcul de la cote

Étant donné les coordonnées  $x$  et  $y$  d'un point de l'espace, sa cote est définie par :

$$z = (x^2 + y^2) \exp\left(-\sqrt{x^2 + y^2}\right) \cos\left(\sqrt{x^2 + y^2}\right)$$



Calculez la cote de chacun des points dans `z1` et `z2` respectivement.

### Orientation

L'expression à calculer contient plusieurs sous-expressions qu'il vaut mieux ne pas recalculer, autant pour accélérer le fonctionnement de l'algorithme que pour simplifier son écriture.

**Outil Java**

Les opérations du cosinus et de l'exponentielle s'écrivent `Math.cos(x)` et `Math.exp(x)`.



Affichez la cote de chacun des points sous la forme suivante :

Cote de (`[x1],[y1]`) est `[z1]`



Testez. Exemple d'exécution :

```
Premier point? -4 3
Deuxieme point? 3.1 4.1
La distance est 7.18471
Cote de (-4,3) est 0.0477825
Cote de (3.1,4.1) est 0.0641817
```

### 1.3 Validation



Validez votre programme avec la solution.

**Solution Java** @[pgdistance1.java]

```
import java.util.Scanner;
import java.util.Locale;

class PGDistance1 {

public static void main(String[] args)
{
    Scanner input = new Scanner(System.in);
    input.useLocale(Locale.US);
    double x1, y1;
    System.out.print("Premier point? ");
    x1 = input.nextDouble();
    y1 = input.nextDouble();
    double x2, y2;
    System.out.print("Deuxieme point? ");
    x2 = input.nextDouble();
    y2 = input.nextDouble();
    double dist = Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
    System.out.println("La distance est " + dist);
    double sc1 = x1 * x1 + y1 * y1;
    double module1 = Math.sqrt(sc1);
    double z1 = sc1 * Math.exp(-module1) * Math.cos(module1);
    System.out.println("Cote de (" + x1 + ", " + y1 + ") est " + z1);
    double sc2 = x2 * x2 + y2 * y2;
    double module2 = Math.sqrt(sc2);
    double z2 = sc2 * Math.exp(-module2) * Math.cos(module2);
    System.out.println("Cote de (" + x2 + ", " + y2 + ") est " + z2);
}
}
```

## **2 Références générales**

**Comprend** [Maysonave-AL1 :c2] ■