

Distance entre deux points [bs10] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel  algoprog  Version 13 mai 2018

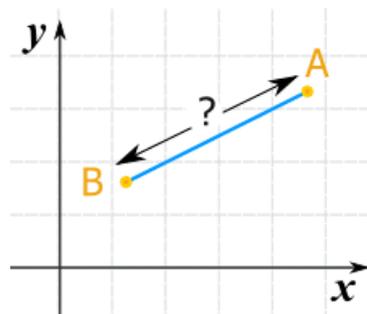
Table des matières

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

C++ - 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 C++

Il n'existe pas de fonction carré (utilisez l'opérateur produit `*` ou la fonction `pow(x,n)` de x^n définie dans la bibliothèque `<cmath>`). La fonction `sqrt(x)` de \sqrt{x} est définie dans la bibliothèque `<cmath>`.



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 C++

Les fonctions `cos(x)` du cosinus et `exp(x)` de l'exponentielle sont également définies dans la bibliothèque `<cmath>`.



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 C++ @[pgdistance1.cpp]

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    double x1, y1;
    cout<<"Premier point? ";
    cin>>x1>>y1;
    double x2, y2;
    cout<<"Deuxieme point? ";
    cin>>x2>>y2;
    double dist = sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
    cout<<"La distance est "<<dist<<endl;
    double sc1 = x1 * x1 + y1 * y1;
    double module1 = sqrt(sc1);
    double z1 = sc1 * exp(-module1) * cos(module1);
    cout<<"Cote de ("<<x1<<","<<y1<<") est "<<z1<<endl;
    double sc2 = x2 * x2 + y2 * y2;
    double module2 = sqrt(sc2);
    double z2 = sc2 * exp(-module2) * cos(module2);
    cout<<"Cote de ("<<x2<<","<<y2<<") est "<<z2<<endl;
}
```

2 Références générales

Comprend [Maysonave-AL1 :c2] ■