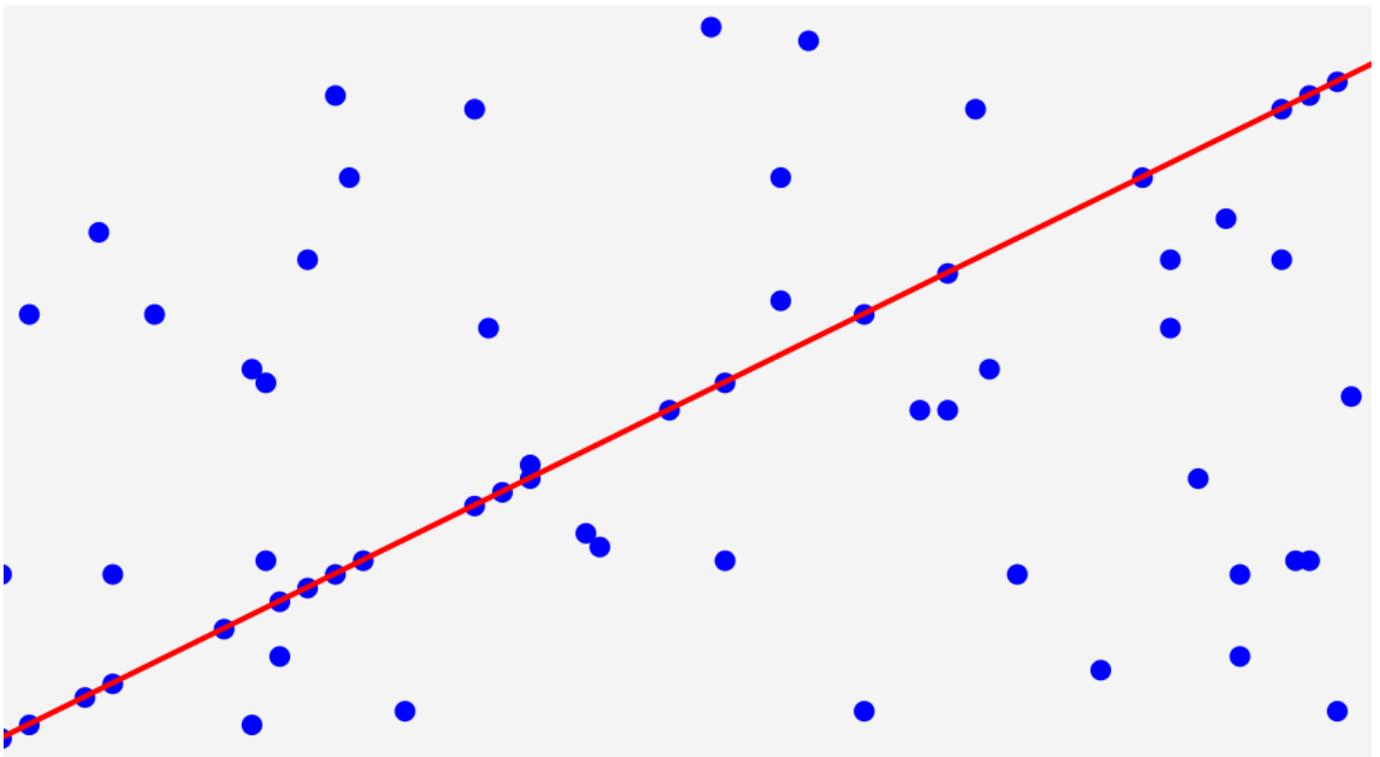
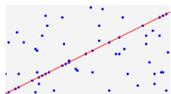


14^e édition
Le 23 mars 2018

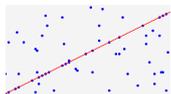


Vous avez dit « online » ?



Règlement

- Date et Lieu :** La 14^{ème} édition du concours de programmation de l'EPFC se déroulera le **vendredi 23 mars 2018** dans les locaux de l'**EPFC** (Avenue de l'Astronomie, 19 à 1210 Bruxelles – **locaux B109 et B113**) de 15h00 à 21h30.
- Candidats :** Le concours est ouvert aux étudiants en deuxième et troisième année du bachelier en informatique de l'**EPFC**, ainsi qu'aux anciens étudiants. Des dérogations pour d'autres étudiants peuvent être accordées. Afin de faciliter l'organisation, des préinscriptions auprès de A. SILOVY (asilovy@ulb.ac.be) sont souhaitées. Les étudiants peuvent concourir en équipe de 2 ou 3 à condition que ces équipes soient clairement identifiées, auprès des responsables, au départ de l'épreuve. Bien entendu, dans ce cas, un éventuel prix sera partagé par les membres de l'équipe.
- Epreuve :** Programmation d'une (ou plusieurs) application(s) console. Tout langage de programmation approprié à l'écriture d'une telle application est accepté. En particulier, l'infrastructure de l'EPFC prévoit l'usage des langages *Java*, *C/C++*, *C#*, *Python* et *Haskell*. A l'issue du concours, les candidats devront remettre le code source de leur solution (ainsi que l'exécutable) sur le serveur du réseau informatique de l'**EPFC**.
- Conditions :** Les participants peuvent apporter toute documentation écrite ou électronique qu'ils peuvent juger utile. Ils peuvent donc employer des bibliothèques de code sur clé *USB*... Cependant, durant la durée de l'épreuve, ils ne peuvent pas communiquer avec l'"extérieur". L'usage d'*Internet* et du *GSM* (à l'exception évidente d'une impérieuse exigence) est proscrit.
- Classement :** Contrairement aux habitudes pédagogiques, le premier critère de sélection sera le fonctionnement des exécutables répondants aux demandes formulées dans l'énoncé. Il correspond donc à **la correction du programme**. Pour départager les candidats, le classement se fera ensuite sur base des critères suivants :
- L'algorithme choisi (et sa complexité)
 - Le choix des structures de données
 - L'analyse du problème
 - L'élégance du code source
 - L'efficacité de la solution
- Prix :** **Des prix sont prévus, soit sous forme de matériel, soit sous forme de bons d'achat.**
- Proclamation :** La proclamation des résultats avec remise des prix sera organisée le **24 avril 2018**. Elle sera suivie d'un drink offert à tous les participants



Le problème : Etant donné un ensemble de N points (x, y) dans le plan discret (x et y sont des entiers positifs ou nuls), nous voulons déterminer si un certain pourcentage de ces points sont alignés.

Les coordonnées des points sont données et un entier P pouvant aller de 20 à 100 vous est donné aussi. Il faut alors déterminer s'il existe oui ou non au minimum P pourcents des points qui sont alignés.

Comment déterminer si des points sont alignés ?

Afin que l'aspect mathématique ne soit pas un obstacle, vous trouverez comment déterminer si 3 points sont alignés en annexe.

Fonctionnement de votre programme

En pratique, votre programme lira sur l'entrée standard un premier nombre entier N ($1 \leq N \leq 1.000.000$) qui indiquera le nombre de points. Votre programme lira ensuite un nombre entier P ($20 \leq P \leq 100$), le pourcentage de points qui doivent être alignés. Il lira ensuite N lignes, chacune avec deux nombres entiers séparés par un espace (les coordonnées x et y des N points avec $0 \leq x, y \leq 1.000.000.000$).

Votre programme écrira simplement les mots « **OUI** » ou « **NON** » selon qu'il existe ou pas P pourcents des points qui sont alignés.

Consignes

Il est impératif que vous respectiez le format exact des Entrées/Sorties spécifié ci-dessus. Votre affichage doit donc respecter scrupuleusement les spécifications données, à savoir, afficher **uniquement** le mot « **OUI** » ou le mot « **NON** ». En particulier, il ne doit pas afficher de message pour la saisie des données (du genre « entrez des nombres » ...) ni afficher un message du style « pas de points alignés ».

Cette consigne a pour objectif de permettre la mise en œuvre de tests automatiques de votre programme. Veillez à la respecter scrupuleusement. Votre programme pourrait être correct et néanmoins ne pas être validé pour cette raison.

Vous ne devez pas vérifier la saisie de l'utilisateur. Vous pouvez donc supposer que le format spécifié ci-dessus sera toujours respecté, l'utilisateur n'entrant jamais que des nombres entiers.



Par exemple, si votre programme se nomme `online`, son exécution sur la ligne de commande pourrait donner lieu à l'interaction ci-dessous :

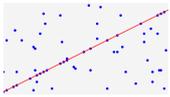
```
C:\>online↵
5↵
60↵
1 0↵
2 2↵
3 4↵
0 1↵
0 2↵
OUI
C:\>online↵
5↵
65↵
1 0↵
2 2↵
3 4↵
0 1↵
0 2↵
NON
C:\>
```

Remarques

Dans cet exemple, la saisie de l'utilisateur est soulignée. Le reste est affiché par l'ordinateur. Remarquez bien : aucunes autres entrée/sortie que celles spécifiées plus haut.

Dans le premier exemple, il faut 60% des points (3 sur les 5 points) alignés. C'est le cas pour les 3 points (1, 0), (2, 2) et (3, 4). La réponse affichée est donc **OUI**. Dans le deuxième exemple, il faut 65% des points, soit au moins 4 sur les 5. Ce n'est pas le cas. La réponse est **NON**.

Faites appel à un professeur pour vous aider si vous ne savez pas comment exécuter votre programme sur la ligne de commande.



Redirection des Entrées/Sorties

Comme vous l'avez étudié par ailleurs, les systèmes d'exploitation permettent une redirection des E/S afin de soulager l'utilisateur de devoir entrer tous les nombres. Votre programme sera exécuté en fournissant des fichiers de données préparés. Donc, par exemple :

```
C:\>online < Exemple.in
OUI
C:\>
```

avec le fichier `Exemple.in` qui contiendrait le texte donné dans le cadre ci-dessous

```
5
60
1 0
2 2
3 4
0 1
0 2
```

Exemple.in

Annexes

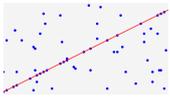
Un programme simple avec un format d'entrées/sorties similaire vous est fourni pour différents langages de programmation (*Java*, *C++*, *C#*, *Python* et *Haskell*). Il vous montre aussi comment déterminer si 3 points sont alignés. **Veillez les utiliser comme modèle.**

Tous les fichiers utiles à votre travail sont disponibles sur le serveur au point **PROFS-PUBLIC\CONCOURS**. Les programmes d'exemples d'entrées/sorties dans les différents langages sont dans le dossier **DemosIO**. Quelques fichiers d'exemples d'input sont dans le dossier **Exemples_Datas**.

Réponses

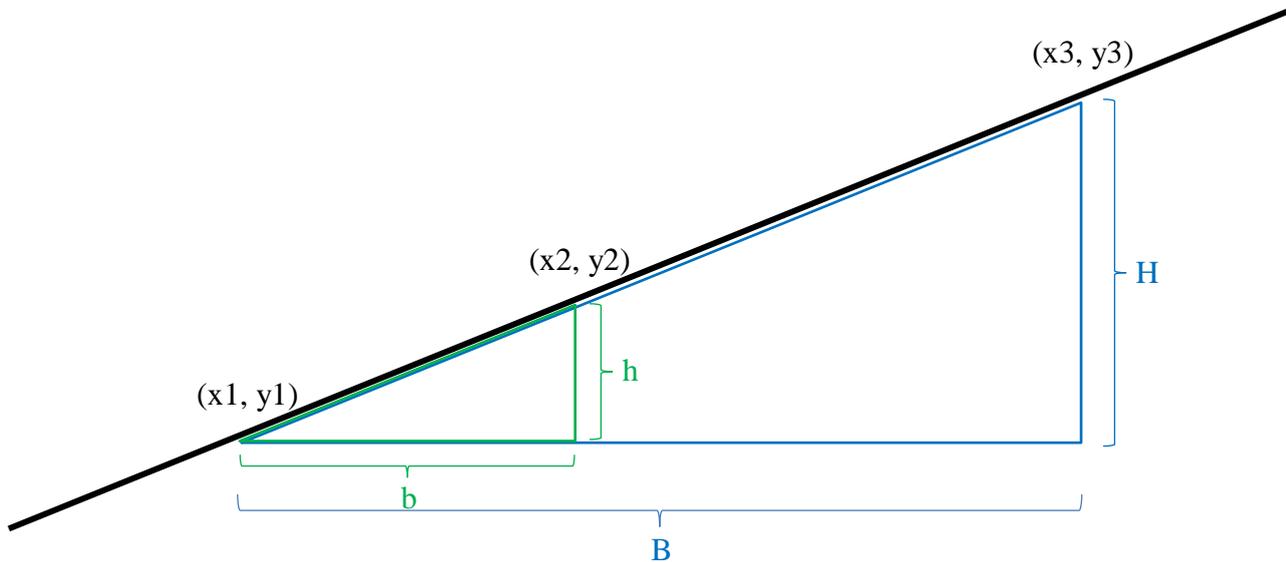
Vous copierez vos solutions (sources et exécutables) dans un répertoire portant votre nom sur le serveur au point **PROFS-PUBLIC\CONCOURS\REPONSES**.

Bon Amusement



Annexes

Une façon de déterminer si trois points sont alignés est de se baser sur la règle des triangles semblables :



Le petit triangle **vert** ayant comme sommets sur la droite les points $(x1, y1)$ et $(x2, y2)$ et le grand triangle **bleu** de sommets $(x1, y1)$ et $(x3, y3)$ sont semblables et le rapport de leurs hauteurs et de leurs bases sont égaux. Donc :

$$\frac{H}{B} = \frac{h}{b}$$

C'est-à-dire :

$$\frac{(y3 - y1)}{(x3 - x1)} = \frac{(y2 - y1)}{(x2 - x1)}$$

Ce qui, pour éviter les divisions sources d'erreurs d'arrondis, peut être réécrit :

$$(y3 - y1) * (x2 - x1) = (x3 - x1) * (y2 - y1)$$



DemoIO.java

```
/*
Programme de démo des E/S pour le concours de programmation EPFC 2018.

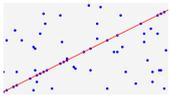
Ce programme lit sur l'entrée standard 3 paires de nombres entiers, deux nombres par
ligne.
Il écrit OUI si les 3 points sont alignés, NON sinon.
*/

import java.util.Scanner;

public class DemoIO {

    // les 3 points sont-ils alignés ?
    // long utilisés pour éviter les dépassements de calculs
    public static boolean aligned3(long x1, long y1,
                                   long x2, long y2,
                                   long x3, long y3) {
        return (y3 - y1) * (x2 - x1) == (x3 - x1) * (y2 - y1);
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int x1 = scan.nextInt();
        int y1 = scan.nextInt();
        int x2 = scan.nextInt();
        int y2 = scan.nextInt();
        int x3 = scan.nextInt();
        int y3 = scan.nextInt();
        if(aligned3(x1, y1, x2, y2, x3, y3))
            System.out.println("OUI");
        else
            System.out.println("NON");
    }
}
```



DemoIO.cpp

```
/*
Programme de démo des E/S pour le concours de programmation EPFC 2018.

Ce programme lit sur l'entrée standard 3 paires de nombres entiers, deux nombres par
ligne.
Il écrit OUI si les 3 points sont alignés, NON sinon.
*/

#include <iostream>
#include <vector>

using namespace std;

// les 3 points sont-ils alignés ?
// long long utilisés pour éviter les dépassements de calculs
bool aligned3(long long x1, long long y1,
              long long x2, long long y2,
              long long x3, long long y3) {
    return (y3 - y1) * (x2 - x1) == (x3 - x1) * (y2 - y1);
}

int main() {
    int x1, y1,
        x2, y2,
        x3, y3;
    cin >> x1 >> y1;
    cin >> x2 >> y2;
    cin >> x3 >> y3;
    if(aligned3(x1, y1, x2, y2, x3, y3))
        cout << "OUI";
    else
        cout << "NON";
}
```