

# Lëtzebuenger Informatiksolympiad 2014

## Demi-Finale

### Problème I (25 points)

### Décomposition

#### Tâche

Une fraction  $\frac{1}{x}$  avec  $x > 0$  peut toujours être écrite comme somme de deux fractions  $\frac{1}{x} = \frac{1}{a} + \frac{1}{b}$  où  $a$  et  $b$  sont deux entiers positifs non nuls avec  $a \leq b$ .

Écrire un programme qui calcule pour un  $x$  donné, combien de telles décompositions sont possibles et qui les affiche toutes.



#### Restrictions

$x$  est un entier tel que  $1 \leq x \leq 10000$ .

#### Entrée et sortie du programme

##### Entrée

- Le programme lit l'entier  $x$ .

##### Sortie

- Le programme affiche sur une ligne le nombre  $d$  de décompositions possibles pour la valeur de  $x$  entrée. Ce nombre est suivi d'une liste triée par valeurs de  $a$  croissantes de  $d$  lignes contenant chacune une décomposition de la fraction  $\frac{1}{x} = \frac{1}{a} + \frac{1}{b}$  en respectant le format donné dans les exemples d'exécution suivants.

#### Exemples d'exécution

##### Exemple d'exécution 1

###### Entrée

5

###### Sortie

2

$1/5 = 1/6 + 1/30$

$1/5 = 1/10 + 1/10$

##### Exemple d'exécution 2

###### Entrée

1001

## Sortie

14

$$1/1001 = 1/1002 + 1/1003002$$

$$1/1001 = 1/1008 + 1/144144$$

$$1/1001 = 1/1012 + 1/92092$$

$$1/1001 = 1/1014 + 1/78078$$

$$1/1001 = 1/1050 + 1/150150$$

$$1/1001 = 1/1078 + 1/14014$$

$$1/1001 = 1/1092 + 1/12012$$

$$1/1001 = 1/1122 + 1/102102$$

$$1/1001 = 1/1144 + 1/8008$$

$$1/1001 = 1/1170 + 1/90090$$

$$1/1001 = 1/1540 + 1/20020$$

$$1/1001 = 1/1638 + 1/18018$$

$$1/1001 = 1/1848 + 1/24024$$

$$1/1001 = 1/2002 + 1/2002$$

**Remettez le programme source sous le nom DECOMPOSITION.xxx, avec xxx=PAS ou C(PP) ou JAVA.**

**Remettez aussi le programme exécutable sous le nom DECOMPOSITION.EXE.**

## Problème II (25 points)

## Logo Lycée

### Tâche

Un lycée cherche un couple de couleurs pour réaliser son logo. La lisibilité du logo est considérée comme essentielle et les deux couleurs devraient donc garantir un contraste important.

Il s'agit d'écrire un programme qui permet de lire un nombre  $E$  d'ensembles de  $C$  couleurs et pour chaque ensemble de déterminer le (ou les) couple(s) offrant un contraste maximal.

Les couleurs sont représentées au format RGB (Red Green Blue) par des triplets de nombres  $(R, G, B)$ , sachant que les valeurs limites sont les suivantes :  $0 \leq R \leq 255, 0 \leq G \leq 255, 0 \leq B \leq 255$ .

Exemples de couleurs :  $(0, 0, 0)$  correspond à la couleur noire ;  $(255, 255, 255)$  correspond à la couleur blanche.

Le contraste entre deux couleurs se mesure par utilisation de la formule suivante :

$$\sqrt{(R1 - R2)^2 + (G1 - G2)^2 + (B1 - B2)^2}$$

sachant que  $(R1, G1, B1)$  est le triplet de nombres de la première couleur, tandis que  $(R2, G2, B2)$  est le triplet de nombres de la deuxième couleur. Pour déterminer les couples offrant un contraste maximal, le résultat donné par la formule ci-dessus est arrondi à l'entier en appliquant l'arrondi usuel.

### Restrictions

Les restrictions suivantes sont à considérer :

- le nombre d'ensembles de couleurs  $E$  :  $1 \leq E \leq 100$  ;
- le nombre de couleurs  $C$  faisant partie d'un ensemble :  $2 \leq C \leq 200$ .

## Entrée du programme

### Principe

La première ligne contient un entier qui représente  $E$ , le nombre de cas (ensembles de couleur) à traiter.

Chaque cas débute par une ligne contenant un entier désignant le nombre de triplets  $C$ . Chaque triplet occupe une ligne contenant trois entiers compris entre 0 et 255 séparés par un espace.

### Exemple

Le programme traitera trois ensembles de couleurs, contenant respectivement cinq, quatre et sept couleurs.

```
3
5
45 254 200
125 59 45
8 252 195
86 16 131
108 160 235
4
0 0 0
0 255 255
255 0 0
255 255 255
7
121 100 67
150 255 177
179 207 202
104 225 16
155 63 234
183 138 57
203 205 131
```

Vous disposez de cet exemple dans le fichier **LogoLycéeInputExempleQuestionnaire.txt**.

Vous disposez d'un deuxième exemple dans le fichier **LogoLycéeInput.txt**.

## Sortie du programme

Pour chaque cas (ensemble de couleurs) le programme affiche :

- dans une première ligne le numéro de l'ensemble de couleurs ;
- dans une deuxième ligne le contraste maximal calculé ;
- dans chacune des  $N$  lignes suivantes ( $N$  étant le nombre de couples offrant un contraste maximal) :
  - les nombres correspondant aux valeurs RGB de la première couleur du couple, séparés par des espaces ;
  - un point-virgule ;
  - les nombres correspondant aux valeurs RGB de la deuxième couleur du couple, séparés par des espaces.

**Exemple (correspondant aux données de l'exemple ci-dessus)**

```

1
270
125 59 45;8 252 195
2
441
0 0 0;255 255 255
0 255 255;255 0 0
3
277
155 63 234;203 205 131

```

Vous disposez du résultat du deuxième exemple dans le fichier **LogoLycéeOutput.txt**.

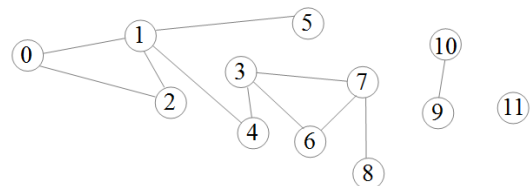
**Remettez le programme source sous le nom LOGOLYCEE.xxx, avec xxx=PAS ou C(PP) ou JAVA.**  
**Remettez aussi le programme exécutable sous le nom LOGOLYCEE.EXE.**

**Problème III (25 points)****Amitiés****Tâche**

Dans un lycée les informations circulent à travers des élèves qui sont amis et forment ainsi des groupes. Une information n'est transmise d'un élève A à un élève B que si A et B sont amis. Grâce à ce comportement de groupe, une information connue par un élève du lycée peut être transmise depuis cet élève à éventuellement tous les autres élèves du lycée. On dispose d'une liste de tous les élèves avec pour chacun la liste de ses amis au lycée. Nous sommes intéressés de connaître tous les liens d'amitié critiques, c'est-à-dire les liens, qui, s'ils étaient rompus, provoqueraient une rupture d'un groupe en deux groupes disjoints.

**Exemple**

Dans cet ensemble de 12 élèves répartis en 3 groupes, les liens 1-4, 1-5, 3-4, 7-8, 9-10 sont des liens d'amitié critiques.

**Restrictions**

Le nombre d'élèves maximal est un entier  $N$  tel que  $1 \leq N \leq 1000$ .

Chacun des  $k$  élèves,  $1 \leq k \leq N$ , est identifié par un numéro unique  $i$  compris entre  $0 \leq i \leq k-1$ .

## Entrée et sortie du programme

### Entrée

- Le programme lit des lignes contenant des entiers positifs. Dans une ligne, le premier entier désigne le numéro d'un élève, les entiers séparés par un espace qui suivent éventuellement sur la même ligne sont les numéros de ses amis. La dernière ligne contient uniquement la valeur -1 qui indique la fin de la liste. Les élèves ne sont pas nécessairement entrés dans l'ordre de leur numéro.

### Sortie

- Le programme affiche tous les couples représentant une amitié critique. Chaque couple est présenté sur une ligne avec les deux numéros d'élèves séparés par un espace. Cette liste éventuellement vide est présentée dans l'ordre croissant du premier numéro d'élève.

## Exemple d'exécution

Pour l'exemple ci-dessus, les données sont à entrer de la manière suivante :

```
0 1 2
1 0 2 5 4
2 0 1
3 4 6 7
4 1 3
5 1
8 7
9 10
10 9
11
6 3 7
7 3 6 8
-1
```

Voici le résultat du programme :

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

**Remettez le programme source sous le nom AMITIES.xxx, avec xxx=PAS ou C(PP) ou JAVA.**

**Remettez aussi le programme exécutable sous le nom AMITIES.EXE.**