

**Le Centre d'éducation
en mathématiques
et en informatique**

*Concours
canadien
d'informatique
2014*

*Niveau
Supérieur*

Commanditaire :

**WATERLOO
MATHEMATICS**

Concours canadien d'informatique — niveau supérieur

Directives à l'intention des participantes et des participants

1. Vous pouvez participer à un concours seulement. Pour participer au concours de niveau intermédiaire, il faut choisir l'autre trousse de problèmes.
2. Sur le formulaire **Information à l'intention des élèves**, indiquez que vous participez au concours de niveau **supérieur**.
3. Vous avez trois (3) heures pour accomplir le travail.
4. Remarquez que :
 - si votre enseignant note votre copie, toutes les entrées se font par le biais d'un fichier nommé `sX.in`, X étant le numéro du problème ($1 \leq X \leq 5$);
 - si vous utilisez le correcteur CCC en ligne (On-line CCC grader), toutes les entrées se font par entrées standard ;
 - toutes les sorties se font par sorties standard, (c.-à-d. par l'écran).

Aucune sollicitation n'est requise. Les sorties doivent être IDENTIQUES à celles des exemples de sorties, par rapport à l'ordre, aux espaces, etc.
5. Vous devez faire votre propre travail. Les tricheurs seront punis sévèrement.
6. Il est interdit de faire appel à des caractéristiques auxquelles le juge (votre enseignant ou le correcteur CCC en ligne) n'a pas accès pendant l'évaluation de votre programme. Si vous utilisez le correcteur CCC en ligne, inscrivez bien le type et la version du compilateur utilisé par votre langage de programmation.
7. Vous pouvez consulter des livres et du matériel écrit. Tout matériel susceptible d'être lu électroniquement (par exemple un programme que vous avez écrit) est *interdit*. Cependant, vous pouvez faire appel aux bibliothèques reconnues pour vos langages de programmation ; par exemple, STL pour C++, `java.util.*`, `java.io.*` et autres pour Java, et ainsi de suite.
8. Vous devez vous limiter aux applications de programmation ordinaires (éditeurs, compilateurs, débogueurs). Toutes les autres applications sont **interdites**. Leur utilisation entraînera une disqualification.
9. Utilisez des noms de fichiers qui sont propres à chaque problème : `s1.pas` ou `s1.c` ou `s1.java` (ou tout autre suffixe de fichier approprié) pour le problème S1. Si vous utilisez le correcteur CCC en ligne, suivez les directives qui y sont données quant aux noms de fichiers, tout en portant une attention particulière aux noms de fichiers pour les programmes Java.
10. Votre programme sera exécuté avec des scénarios d'essais différents de ceux qui figurent comme exemples. Assurez-vous de vérifier votre programme au moyen d'autres scénarios d'essais. Assurez-vous d'avoir un code aussi performant que possible par rapport au temps.

Les solutions inefficaces peuvent faire perdre des points, particulièrement dans les problèmes 4 et 5.

11. Dans ce concours de niveau supérieur, vous disposerez d'au plus 5 secondes d'exécution par scénario d'essai, que la notation soit faite par l'enseignant ou de façon électronique par le correcteur en ligne. Aucun point ne sera accordé pour un scénario d'essai dont l'exécution prend plus de 5 secondes.
12. Si vous vous placez parmi les 20 premiers participants et participantes dans le concours du niveau supérieur, vous serez invité à participer à l'Étape 2 du CCI, qui aura lieu à l'Université de Waterloo au mois de mai 2014. Si vous vous placez parmi les premiers lors de l'Étape 2, vous serez invité à participer à l'équipe qui représentera le Canada à IOI 2014, à Taiwan. Notez que vous devez connaître C, C++ ou Pascal si vous êtes invité à l'Étape 2. Mais d'abord, vous devez réussir le concours d'aujourd'hui !
13. Consultez le site web du CCI à la fin du mois de mars pour connaître votre classement dans ce concours et pour connaître le nom des gagnants. Voici l'adresse :

www.cemc.uwaterloo.ca/coc

Problème S1 : Invitation à une fête

Description du problème

Vous préparez une fête, mais vous n'avez pas assez de place pour inviter tous vos amis. Pour déterminer les amis que vous inviterez, vous utilisez la méthode suivante qui est exempte d'émotions.

Attribuez un numéro $1, 2, \dots, K$ à chacun de vos K amis et placez les numéros dans une liste en ordre croissant. Il y aura ensuite m rondes. Dans chaque ronde, vous utilisez un nombre pour déterminer quels numéros d'amis seront enlevés de la liste.

Dans les rondes, les nombres r_1, r_2, \dots, r_m seront utilisés. Dans la ronde i , vous enlevez les numéros d'amis dont la position actuelle en liste est un multiple de r_i (c'est-à-dire les positions $r_i, 2r_i, 3r_i, \dots$). Dans chaque ronde, la nouvelle liste commence par la position 1.

Votre programme donnera en sortie les numéros d'amis qui sont encore en liste à la fin de toutes les rondes.

Précisions par rapport aux entrées

La première ligne d'entrée contient un entier K ($1 \leq K \leq 100$). La deuxième ligne contient un entier m ($1 \leq m \leq 10$), soit le nombre de rondes. Chacune des m lignes suivantes contient un entier. La $i^{\text{ème}}$ de ces lignes ($1 \leq i \leq m$) contient r_i ($2 \leq r_i \leq 100$), ce qui indique que les numéros d'amis dont la position est un multiple de r_i doivent être enlevés de la liste.

Précisions par rapport aux sorties

La sortie contient les numéros d'amis qui n'auront pas été enlevés. Ils doivent paraître en ordre croissant, un numéro par ligne.

Exemple d'entrée

```
10
2
2
3
```

Sortie pour l'exemple d'entrée

```
1
3
7
9
```

Explication de la sortie pour l'exemple d'entrée

Au départ, les numéros d'amis sont 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Il y a deux rondes. Dans la première ronde, on enlève les numéros dans les positions paires (les positions 2, 4, 6, 8 et 10) et à la fin de cette ronde, la liste comporte les numéros d'amis 1, 3, 5, 7, 9. Dans la deuxième ronde, on enlève les numéros dans les positions qui sont des multiples de 3 (la position 3 qui contient le numéro 5). À la fin de cette ronde, la liste comporte les numéros d'amis 1, 3, 7, 9.

Problème S2 : Attribution des partenaires

Description du problème

Le CEMI organise un atelier. Dans une activité, les participants doivent travailler en paires. Le comité organisateur décide de choisir les partenaires maintenant, avant le début de l'atelier. Vous devez déterminer si le choix de partenaires est *cohérent*, c'est-à-dire que si A est le partenaire de B, alors B doit être le partenaire de A et aucun participant n'est son propre partenaire.

Précisions par rapport aux entrées

L'entrée comporte trois lignes. La première contient un entier N ($1 < N \leq 30$), soit le nombre d'élèves inscrits. La deuxième ligne contient le prénom des N élèves, séparés d'une espace. (Les noms ne contiennent que des lettres majuscules ou minuscules et tous les prénoms sont distincts.) La troisième ligne contient les mêmes N prénoms, dans un certain ordre, séparés d'une espace.

Les positions des noms dans les deux dernières lignes indiquent les partenaires : le $i^{\text{ième}}$ nom dans la deuxième ligne est le partenaire du $i^{\text{ième}}$ nom dans la troisième ligne.

Précisions par rapport aux sorties

Si la notation est faite par l'enseignant, la sortie est `bon` si les deux listes donnent un choix de partenaires cohérent et `mal` si elles donnent un choix qui n'est pas cohérent.

Si la notation est faite par le correcteur en ligne, la sortie est `good` si les deux listes donnent un choix de partenaires cohérent et `bad` si elles donnent un choix qui n'est pas cohérent.

Exemple d'entrée 1

4

Ada Alan Grace John
John Grace Alan Ada

Sortie pour l'exemple d'entrée 1

`bon` (si noté par l'enseignant) ou `good` (si noté par le correcteur en ligne)

Explication de la sortie pour l'exemple d'entrée 1

Ada et John sont partenaires ; Alan et Grace sont partenaires. Le choix est cohérent.

Exemple d'entrée 2

7

Rich Graeme Michelle Sandy Vlado Ron Jacob
Ron Vlado Sandy Michelle Rich Graeme Jacob

Sortie pour l'exemple d'entrée 2

`mal` (si noté par l'enseignant) ou `bad` (si noté par le correcteur en ligne)

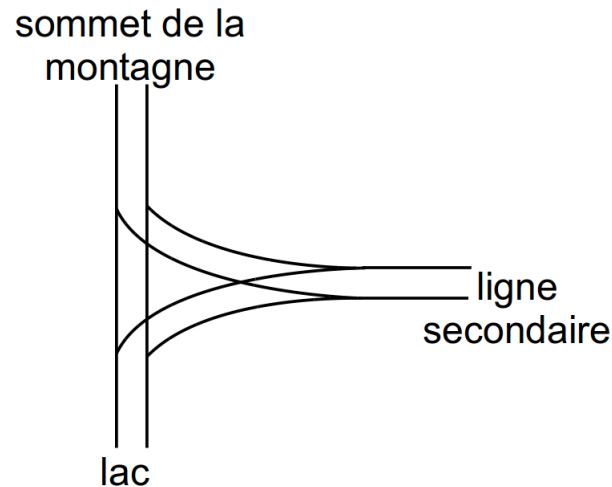
Explication de la sortie pour l'exemple d'entrée 2

Graeme a pour partenaire Vlado, mais Vlado a pour partenaire Rich. Le choix n'est pas cohérent. De plus, Jacob est son propre partenaire, ce qui est incohérent.

Problème S3 : La confection de Genève

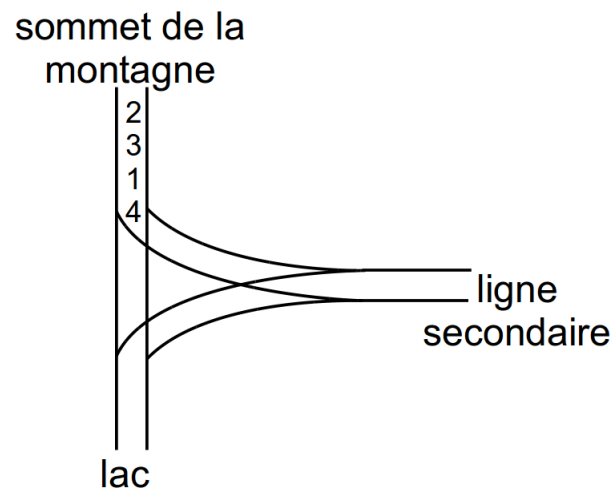
Description du problème

Pour assurer la paix et la prospérité aux générations futures, l'Organisation des nations unies s'apprête à confectionner le plus grand bonbon du monde. Les ingrédients doivent être transportés par wagons de trains du sommet d'une montagne et versés dans le lac de Genève. Le chemin de fer descend une pente raide et il comporte aussi une ligne secondaire, comme dans la figure suivante.



À l'heure actuelle, chacun des N ingrédients se trouve dans son propre wagon. Chacun des wagons a son propre numéro de 1 à N . Les ingrédients doivent être versés dans le lac dans l'ordre 1, 2, 3, ..., N , mais les wagons sont alignés dans un ordre quelconque. Puisque les wagons et leur marchandise sont très lourds, il est seulement possible de les tirer en descendant vers le lac ou vers la ligne secondaire. Est-il possible de verser les ingrédients dans le lac dans l'ordre 1, 2, 3, ..., N ?

Par exemple, si les wagons étaient dans l'ordre 2, 3, 1, 4, comme dans la figure suivante, on pourrait verser les ingrédients dans le lac dans le bon ordre en procédant comme suit :



- Tirer le wagon 4 sur la ligne secondaire.
- Tirer le wagon 1 et verser son contenu dans le lac.
- Tirer le wagon 3 sur la ligne secondaire.
- Tirer le wagon 2 et verser son contenu dans le lac.
- Tirer le wagon 3 de la ligne secondaire vers le lac et verser son contenu dans le lac.
- Tirer le wagon 4 de la ligne secondaire vers le lac et verser son contenu dans le lac.

Précisions par rapport aux entrées

La première ligne contient le nombre E ($1 \leq E \leq 10$), soit le nombre d'essais qui seront présentés. Elle est suivie des essais. Chaque essai contient un entier N ($1 \leq N \leq 100\,000$) sur une ligne, suivi des numéros de N wagons, une par ligne, tels qu'ils sont alignés du haut vers le bas. Les numéros de wagons seront toujours les numéros de 1 à N présentés dans un ordre quelconque.

Précisions par rapport aux sorties

Pour chaque essai, la sortie est Y (pour « youpi ») si la recette peut être complétée ou N si elle ne peut pas être complétée.

Exemple d'entrée

2
4
2
3
1
4
4
4
1
3
2

Sortie pour l'exemple d'entrée

Y
N

Problème S4 : Vitrail

Description du problème

Vous placez N morceaux rectangulaires de verre gris pour former un vitrail. On a attribué à chaque morceau de verre un numéro qui représente son « degré de gris ». Lorsque deux morceaux de verre sont superposés, le degré de gris obtenu est la somme des degrés de gris des morceaux.

Vous connaissez la position voulue de chaque morceau de verre et chaque morceau doit être placé de manière que ses côtés soient parallèles à l'axe des abscisses ou à l'axe des ordonnées (aucun morceau ne peut donc être en position oblique).

Vous voulez connaître l'aire totale des parties du vitrail qui ont un degré de gris supérieur ou égal à G .

Précisions par rapport aux entrées

La première ligne contient un entier N ($1 \leq N \leq 1000$), soit le nombre de morceaux de verre. La deuxième ligne contient l'entier G ($1 \leq G \leq 1\,000\,000\,000$), soit le seuil à atteindre pour le degré de gris. Chacune des N lignes suivantes contient cinq entiers, représentant la position du coin supérieur gauche et du coin inférieur droit du $i^{\text{ième}}$ morceau de verre, suivie de son degré de gris. De façon plus précise, les cinq entiers sont présentés dans l'ordre $x_g y_h x_d y_b g_i$, le coin supérieur gauche étant situé au point (x_g, y_h) , le coin inférieur droit étant situé au point (x_d, y_b) et g_i étant le degré de gris. Vous pouvez supposer que $1 \leq g_i \leq 1\,000\,000$. Le coin inférieur gauche ne peut être placé à la gauche ou en dessous du point $(0, 0)$. Vous pouvez supposer que $0 \leq x_g < x_d \leq K$ et que $0 < y_h < y_b \leq K$.

De plus, les conditions suivantes seront en vigueur :

- Au moins 10 % des points seront attribués pour des scénarios d'essais où $N \leq 100$ et $K \leq 100$;
- au moins 30 % des points seront attribués pour des scénarios d'essais où $N \leq 1000$ et $K \leq 1000$;
- au moins 40 % des points seront attribués pour des scénarios d'essais où $N \leq 100$ et $K \leq 1\,000\,000\,000$;
- les autres points seront attribués pour des scénarios d'essais où $N \leq 1000$ et $K \leq 1\,000\,000\,000$.

Précisions par rapport aux sorties

La sortie est l'aire totale des parties du vitrail qui ont un degré de gris supérieur ou égal à G . Chaque sortie sera inférieure à 2^{64} . Dans certains scénarios, la sortie sera supérieure à 2^{32} .

Exemple d'entrée

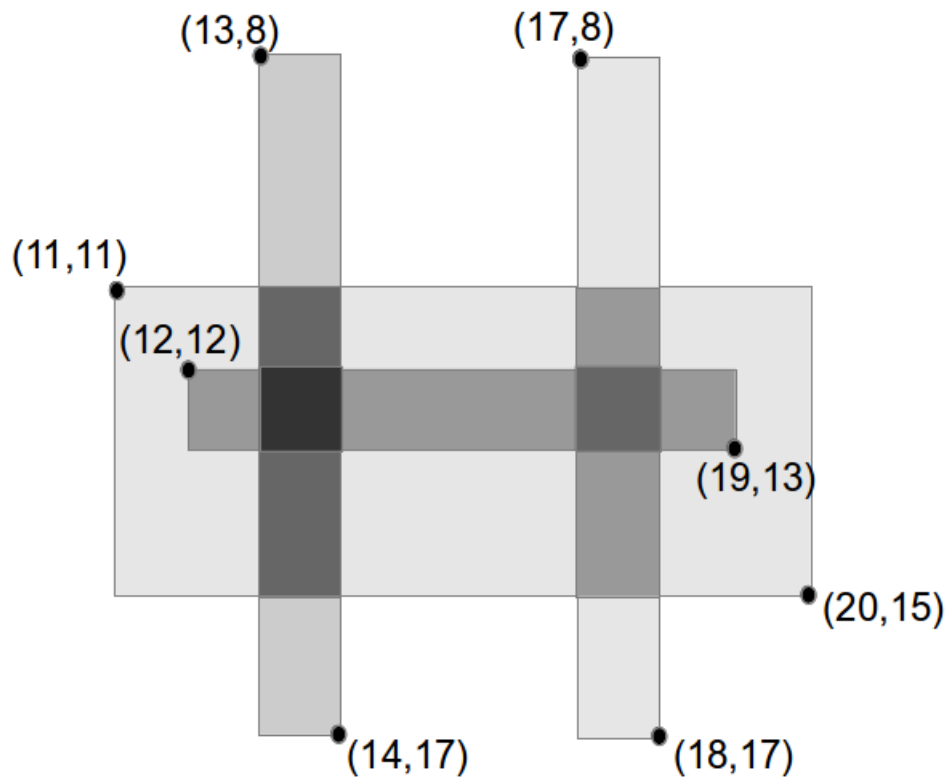
```
4
3
11 11 20 15 1
13 8 14 17 2
17 8 18 17 1
12 12 19 13 1
```

Sortie pour l'exemple d'entrée

5

Explication de la sortie pour l'exemple d'entrée

Quatre morceaux de verre sont utilisés. Deux régions ont un degré de gris supérieur à 3 : la première est située entre les points (13, 11) et (14, 15) (elle a un degré de gris égal à 3, sauf pour un carré-unité qui a un degré de gris égal à 4) ; la deuxième est située entre les points (17, 12) et (18, 13) (elle a un degré de gris égal à 3). Comme on le voit dans la figure ci-dessous, ces deux régions ont une aire totale de 5 et un degré de gris supérieur ou égal à 3.



Problème S5 : Le renard paresseux

Description du problème

Votre animal de compagnie est un renard. Il aime les gâteaux. Vous avez N voisins, situés à divers endroits (représentés par des points dans le plan cartésien), qui remettent des gâteaux à votre renard et chaque voisin a un nombre illimité de gâteaux à donner. Le renard commence à l'origine qui ne correspondra pas à un des N endroits où sont situés les voisins.

Que demande le renard pour recevoir les gâteaux ? Bonne question, mais cela ne nous concerne pas.

Le renard se déplace d'un endroit à un autre et reçoit exactement une gâteaux à chaque visite. Il peut revisiter le même endroit, mais ne peut pas faire deux visites *consécutives* au même endroit.

Votre renard est paresseux. La distance qu'il est prêt à parcourir d'un endroit à un autre est strictement décroissante. Ainsi la distance de l'origine jusqu'au premier endroit choisi doit être supérieure à la distance du premier endroit jusqu'au deuxième endroit ; cette distance doit être supérieure à la distance du deuxième endroit jusqu'au troisième endroit et ainsi de suite.

Quel est le plus grand nombre de gâteaux que le renard peut recevoir ?

Précisions par rapport aux entrées

La première ligne contient l'entier N ($1 \leq N \leq 2000$). Les N lignes suivantes contiennent chacune X_i , suivi d'une espace, suivie de Y_i , avec $i = 1, 2, \dots, N$ ($-10\,000 \leq X_i, Y_i \leq 10\,000$) représentant les coordonnées de la $i^{\text{ème}}$ position.

De plus, les conditions suivantes seront en vigueur :

- Au moins 20 % des points seront attribués pour des scénarios d'essais où $N \leq 50$;
- au moins 40 % des points seront attribués pour des scénarios d'essais où $N \leq 200$;
- les autres points seront attribués pour des scénarios d'essais où $N \leq 2000$.

Précisions par rapport aux sorties

La sortie est un entier, soit le plus grand nombre de gâteaux que le renard peut recevoir.

Exemple d'entrée

```
5
5 8
4 10
3 1
3 2
3 3
```

Sortie pour l'exemple d'entrée

```
6
```

Explication de la sortie pour l'exemple d'entrée

Le renard visite les endroits suivants dans l'ordre (les distances sont données) :

- De $(0, 0)$ à $(4, 10)$, une distance de $\sqrt{116}$
- De $(4, 10)$ à $(3, 1)$, une distance de $\sqrt{82}$
- De $(3, 1)$ à $(5, 8)$, une distance de $\sqrt{53}$
- De $(5, 8)$ à $(3, 3)$, une distance de $\sqrt{29}$
- De $(3, 3)$ à $(3, 1)$, une distance de 2
- De $(3, 1)$ à $(3, 2)$, une distance de 1