Le Centre d'éducation en mathématiques et en informatique

Concours canadien d'informatique 2009

Niveau supérieur

Commanditaire :

University of Waterloo

Concours canadien d'informatique – Niveau supérieur

Règles et conseils 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. Vous pouvez prendre pour acquis que :
   - toutes les entrées se trouvent dans des fichiers nommés `sX.in`, X étant le numéro du problème ($1 \leq X \leq 5$). Le fichier d'entrées du Problème S1 est donc `s1.in`.
   - toutes les sorties se font par l'écran.

   Puisque les entrées se trouvent dans des fichiers, il n'y aura aucune sollicitation. 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, n'a pas accès pendant l'évaluation de votre programme.

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.*, etc. 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 fichier qui sont propres à chaque problème : par exemple, `s1.pas` ou `s1.c` ou `s1.java` (ou tout autre suffixe de fichier approprié) pour le problème S1. Ceci facilitera la tâche de l'évaluateur.

10. Votre programme sera exécuté avec des fichiers d'essai différents de ceux qui figurent comme exemples. Assurez-vous de vérifier votre programme au moyen d'autres fichiers d'essai. Pour certains problèmes, particulièrement les problèmes 4 et 5, des solutions peu performantes peuvent faire perdre des points. Assurez-vous d'avoir un code aussi performant que possible par rapport au temps.

11. Les deux premiers participants de chaque région du pays recevront une plaque et une somme de 100 $. Leur école recevra aussi une plaque. Les régions sont :

– L’ouest (de la C.-B. au Manitoba)
– Le nord et l’est de l’Ontatio
– Toronto métropolitain
– Le centre et l’ouest de l’Ontario
– Le Québec et les provinces de l’Atlantique

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 2009. Si vous vous placez parmi les 4 premiers lors de l’Étape 2, vous serez invité à participer à l’équipe qui représentera le Canada à IOI 2009, en Bulgarie. 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, pour voir comment on pouvait résoudre les problèmes et pour connaître le nom des gagnants. Voici l’adresse :

    `www.cemc.uwaterloo.ca/ccc`

Problème S1 : Nombres cool

Description du problème
Éric aime les nombres intéressants comme 64. Or, le nombre 64 est à la fois un carré parfait et un cube parfait, puisque $64 = 8^2$ et $64 = 4^3$. Éric dit qu'un tel nombre est *cool*.

Écrivez un programme qui aide Éric à déterminer combien il y a d'entiers cool dans un intervalle donné.

Précisions par rapport aux entrées
Dans la première ligne d'entrée, vous recevrez un entier a tel que $a \geq 1$ et $a \leq 10^8$. Dans la deuxième ligne d'entrée, vous recevrez un entier b tel que $a \leq b$ et $b \leq 10^8$.

Précisions par rapport aux sorties
La sortie doit indiquer combien il y a de nombres cool dans l'intervalle de a à b (si a et b sont des nombres cool, ils sont comptés dans l'intervalle).

Exemple d'entrée 1
```
1
100
```

Sortie pour l'entrée 1
```
2
```

Exemple d'entrée 2
```
100
1000
```

Sortie pour l'entrée 2
```
1
```

Problème S2 : Lumières éteintes ou allumées

Description du problème
On vous a donné une grille de R rangées de lumières ($1 < R < 30$), chaque rangée contenant L lumières ($1 \leq L < 8$). Chaque lumière peut être allumée ou éteinte. Dans ce problème, la rangée du bas est la rangée 1 et la rangée du haut est la rangée R. À côté de chaque rangée à l'exception de la rangée du haut (c.-à-d. la rangée R), il y a un bouton sur lequel on peut appuyer.

Si on appuie sur le bouton de la rangée k ($1 \leq k < R$), il en résulte une opération sur chaque lumière de la rangée k, tel qu'il est décrit ci-dessous. On considère la colonne i de la rangée k, où $1 \leq i \leq L$. Si la lumière de la colonne i de la rangée k et la lumière de la colonne i de la rangée $k+1$ sont toutes deux allumées ou toutes deux éteintes, alors en appuyant sur le bouton de la rangée k, la lumière de la colonne i de la rangée k sera éteinte. Si une de ces deux lumières est allumée et que l'autre est éteinte, alors en appuyant sur le bouton de la rangée k, la lumière de la colonne i de la rangée k sera allumée. Voici un exemple où $L = 4$:

Numéros de colonne	1	2	3	4
Rangée $k+1$	allumée	allumée	éteinte	éteinte
Rangée k avant d'appuyer sur le bouton	allumée	éteinte	allumée	éteinte
Rangée k après avoir appuyé sur le bouton	éteinte	allumée	allumée	éteinte

On vous dit quelles lumières sont allumées et quelles lumières sont éteintes au départ. Vous devez calculer le nombre possible de dispositions de lumières allumées et éteintes de la rangée du bas résultant de n'importe quelle séquence de boutons sur lesquels on a appuyé.

Description de l'entrée
La première ligne de l'entrée contiendra un entier R qui représente le nombre de rangées. La deuxième ligne de l'entrée contiendra un entier L qui représente le nombre de lumières par ligne. Les R lignes d'entrée suivantes contiendront chacune L entiers, chaque entier étant un 0 (pour indiquer « éteinte ») ou un 1 (pour indiquer « allumée »). Chaque entier sera séparé de l'entier précédent par une espace. Ces R lignes seront données en ordre descendant : la troisième ligne d'entrée décrira la rangée R, la quatrième ligne d'entrée décrira la rangée $R-1$, ainsi de suite, et la dernière ligne d'entrée décrira la rangée du bas.

Description de la sortie
La sortie donnera le nombre de dispositions possibles des lumières de la rangée du bas.

Exemple d'entrée
```
4
3
0 0 1
0 1 1
1 0 1
0 0 1
```

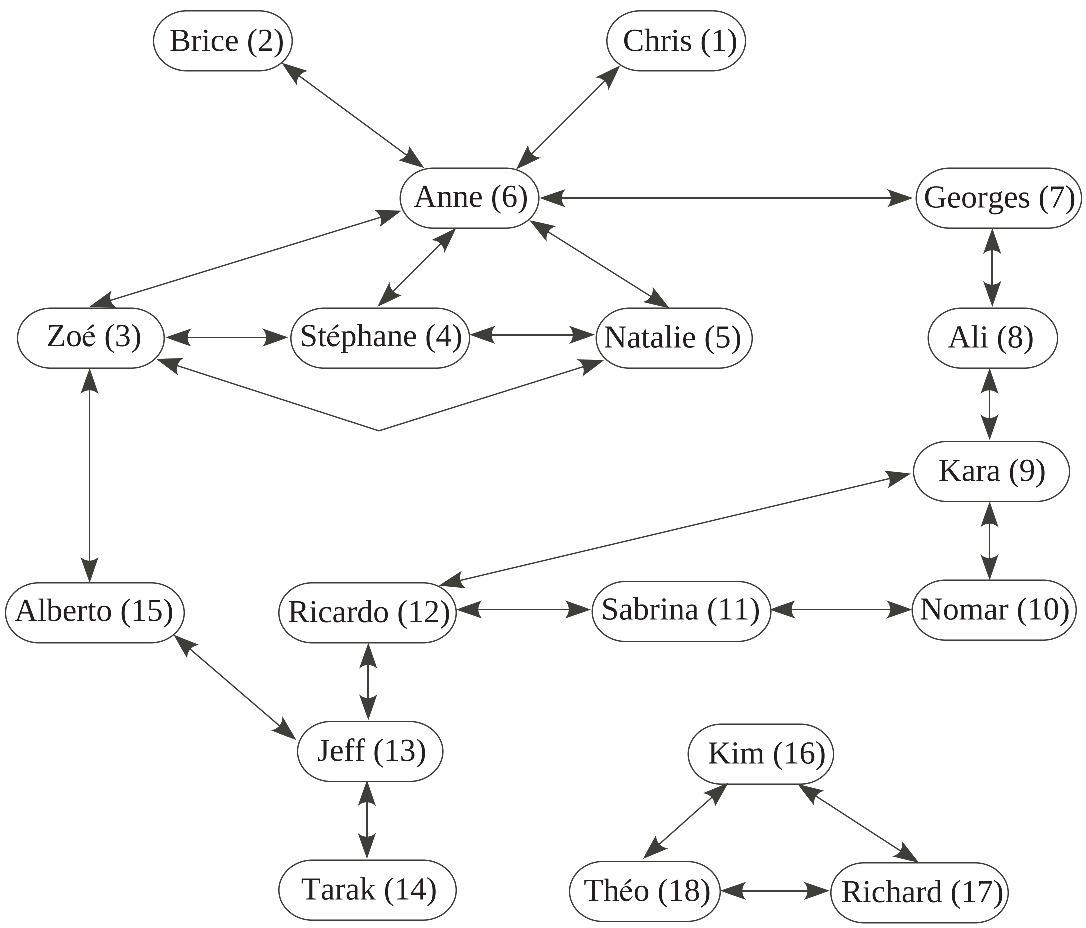
Sortie pour l'exemple
```
4
```

Problème S3 : Degrés de séparation

Description du problème

Facebook est sans doute le principal outil des élèves pour socialiser. Facebook suscite plusieurs questions reliées à l'informatique, telles que le degré de séparation entre deux personnes.

Par exemple, dans la figure ci-dessous, il y a plusieurs chemins qui relient Anne et Alberto. En voici quelques-unes :
– Anne → Zoé → Alberto
– Anne → Natalie → Zoé → Alberto
– Anne → Georges → Ali → Kara → Ricardo → Jeff → Alberto
Le chemin le plus court entre Anne et Alberto comporte deux étapes (Anne → Zoé et Zoé → Alberto). On dit donc qu'il y a 2 degrés de séparation entre eux. On peut aussi dire que Alberto est un ami d'une amie d'Anne.



Vous pouvez supposer qu'au départ, la figure précédente représente bien tous les liens d'amitié entre les personnes. Vous devrez placer ces liens dans votre programme. Or, ces liens peuvent changer et votre programme doit tenir compte de ces changements. Par exemple, des amitiés peuvent commencer, possiblement avec de nouvelles personnes. Des amitiés peuvent se terminer. Votre programme doit être en mesure de trouver les amis des amis et de déterminer les degrés de séparation entre deux amis.

Description des entrées et des sorties

Votre programme doit pouvoir lire six commandes possibles. Chaque commande indique une action particulière (voir ci-dessous). Vous pouvez supposer que x et y sont des entiers, que $x \neq y$, $x \geq 1$, $y \geq 1$, $x < 50$ et $y < 50$. Vous pouvez auss supposer que les commandes (`i`, `d`, `n`, `f`, `s`, `q`) sont présentées une par ligne et que les paramètres (zero, un ou deux nombres entier) paraissent une par ligne.

- `i` x y – Créer un lien d'amitié entre la personne x et la personne y. Si x et y sont déjà amis, ne rien ajouter. Si x ou y est une nouvelle personne, l'ajouter.
- `d` x y – Défaire le lien d'amitié entre la personne x et la personne y.
- `n` x – Sortir le nombre d'amis de la personne x.
- `f` x – Sortir le nombre d'« amis d'amis » de la personne x. Avis que x et les amis directes de x ne sont pas compter pour « amis d'amis ».
- `s` x y – Sortir le nombre de degrés de séparation entre x et y. S'il n'y a aucun chemin entre x et y, sortir `Aucun lien`.
- `q` – Quitter le programme.

Exemple d'une interaction

Entrée	Sortie	Explication

`i`
`20`
`10`
	(aucune sortie)	L'insertion d'un lien d'amitié ne produit aucune sortie.

`i`
`20`
`9`
	(aucune sortie)	L'insertion d'un lien d'amitié ne produit aucune sortie.

`n` `20`	`2`	La personne 20 a deux amis (10 et 9)
`f` `20`	`3`	Les amis des amis de 20 sont 8, 11, 12.

`s`
`20`
`6`
	`4`	Le plus court chemin est $20 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 6$.

`q`	(aucune sortie)	Le programme est terminé.

Problème S4 : Ventes en ligne

Description du problème
Le pays de Doubleclic compte N villes ($N \leq 5000$). Il existe plusieurs routes commerciales entre les villes. En tout, il y a T routes commerciales ($0 \leq T \leq 25\,000\,000$) dans le pays. Étant donné une route commerciale entre deux villes x et y, il y a un coût $C(x, y)$ lié au transport de la marchandise entre les villes, où $C(x, y) \geq 0$, $C(x, y) \leq 10\,000$ et $C(x, y) = C(y, x)$. Parmi les N villes, K d'entre elles ($1 \leq K \leq N$) ont des magasins qui vendent de fort jolis crayons en ligne. Dans la ville x, le coût d'un crayon est de P_x ($0 \leq P_x \leq 10\,000$).

Vous devez déterminer le coût minimal pour acheter et expédier un crayon à la ville D ($1 \leq D \leq N$) en utilisant la séquence de routes commerciales la moins dispendieuse. Remarquez qu'il peut être possible d'acheter le crayon dans la ville D, ce qui élimine le coût d'expédition.

Description de l'entrée
La première ligne d'entrée contient le nombre N de villes. Vous pouvez supposer que les villes sont numérotées de 1 à N. La deuxième ligne d'entrée contient le nombre T de routes commerciales. Chacune des T lignes suivantes contient trois entiers, soit x, y et $C(x, y)$, où $C(x, y)$ indique le coût pour expédier de la marchandise entre les villes x et y. La ligne suivante contient l'entier K qui indique le nombre de villes qui ont un magasin qui vend les crayons en ligne. Chacune des K lignes suivantes contient deux entiers, soit z et P_z, qui indiquent le coût P_z d'un crayon dans la ville z. La dernière ligne contient l'entier D, qui indique la ville où le crayon doit être expédié.

Description de la sortie
La sortie est le coût total minimal pour l'achat en ligne d'un crayon et son expédition vers la ville D.

Exemple d'entrée
```
3
3
1 2 4
2 3 2
1 3 3
3
1 14
2 8
3 3
1
```

Sortie pour l'exemple
```
6
```

Problème S5 : Sans fil

Description du problème
Bob est seul à la maison, assis devant son ordinateur. Il aimerait un peu d'interaction sociale et songe à se rendre à un café-bar avec son ordinateur.

Bob a de nombreuses données sur les cafés-bars et les réseaux sans fil dans sa ville. Dans sa ville, il y a un café-bar à *chaque* carrefour. Dans cette ville, il y a M rues est-ouest ($1 \leq M \leq 30\,000$) et N rues nord-sud ($1 \leq N \leq 1000$). De plus, il y a une distance de 1 mètre entre les rues parallèles consécutives (la ville est très compacte).

Dans K des cafés-bars ($1 \leq K \leq 1000$), il y a une poste de réseau sans fil. Chaque poste a un débit binaire particulier B ($1 \leq B \leq 1000$) et peut être capté dans un rayon de R mètres ($1 \leq R \leq 30\,000$) du café-bar. En d'autres mots, un poste de réseau sans fil situé dans un café-bar forme un cercle de rayon R mètres centré dans ce café-bar. On peut utiliser le poste de réseau sans fil si on est à moins de R mètres du poste et on ne peut pas l'utiliser si on est à plus de R mètres du poste.

Vous pouvez supposer que chaque café-bar a au plus un poste de réseau sans fil, mais que si on est assis dans un café-bar, il peut être possible d'avoir accès à plus d'un poste à cause de leur proximité.

L'ordinateur de Bob est muni d'un dispositif qui lui permet d'utiliser à la fois tous les débits binaires de tous les postes auxquels il peut accéder.

Bob aimerait trouver le débit binaire maximal qu'il peut obtenir et le nombre de cafés-bars qui lui offrent ce débit maximal.

Description de l'entrée
Sur la première ligne d'entrée, vous recevrez un entier M qui représente le nombre de rues est-ouest. Sur la deuxième ligne d'entrée, vous recevrez un entier N qui représente le nombre de rues nord-sud. Sur la troisième ligne d'entrée, vous recevrez un entier K qui représente le nombre de cafés-bars qui ont un poste de réseau sans fil. Sur chacune des K lignes suivantes, il y aura 4 entiers. Le premier entier x d'une de ces lignes représente le numéro de la rue nord-sud où est situé le café-bar, $1 \leq x \leq N$. Le deuxième entier y d'une de ces lignes présente le numéro de la rue est-ouest où est situé le café-bar, $1 \leq y \leq M$. Le troisième entier R d'une de ces lignes présente le rayon du poste de réseau sans fil de ce café-bar. Le quatrième entier B d'une de ces lignes présente le débit binaire du poste de réseau sans fil de ce café-bar.

Description de la sortie
La sortie comprendra deux lignes. La première ligne de sortie sera un entier qui indique le débit binaire maximal qui peut être obtenu parmi tous ces cafés-bars. La deuxième ligne de sortie sera un entier qui indique le nombre de cafés-bars où on peut utiliser ce débit binaire maximal.

Exemple d'entrée

```
3
5
3
1 3 2 5
3 1 2 7
5 1 1 5
```

Sortie pour cet exemple

```
12
5
```

Explication de l'entrée et de la sortie

Dans la figure suivante, on remarque que les 5 cafés-bars qui sont indiqués par un gros point noir ont accès à un débit binaire total de 12.