Le Centre d'éducation en mathématiques et en informatique

Concours canadien d'informatique 2008

Niveau supérieur

Commanditaire :

University of Waterloo

Concours canadien d'informatique

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.

   Dans certains problèmes, on peut vous demander de fournir une sollicitation pour l'utilisateur. Si aucune sollicitation n'est requise, il n'est pas nécessaire d'en fournir une. 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 2008. 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 2008, en Égypte. 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 : Il fait froid !

Description du problème

Il fait froid au Canada en hiver. Votre tâche est très simple : il faut trouver la ville la plus froide au Canada. Donc, étant donné une liste de villes et de leur température, vous devez déterminer la ville de la liste dont la température est la plus basse, c'est-à-dire qui est la plus froide.

Précisions par rapport aux entrées

Les entrées sont une série de lignes. Chaque ligne indique le nom d'une ville et une température. Les températures sont des entiers qui peuvent être précédés d'un signe négatif. Il y a une seule espace entre le nom de la ville et la température. Aucun nom de ville ne contient d'espace et chaque nom a moins de 256 caractères. La liste contient au moins une ville et pas plus de 10 000 villes. La dernière ville est toujours Waterloo. Vous pouvez supposer que les températures ne seront pas sous -273 degrés ou au-dessus de 200 degrés.

Précisions par rapport aux sorties

Sortez le nom de la ville la plus froide, sur une seule ligne, sans espace avant ou après. Vous pouvez supposer qu'il n'y aura pas plus d'une ville la plus froide.

Exemple d'entrée

```
Saskatoon -20
Toronto -2
Winnipeg -40
Vancouver 8
Halifax 0
Montreal -4
Waterloo -3
```

Sortie pour l'exemple

```
Winnipeg
```

Problème S2 : Des sous dans l’anneau

Description du problème
Les programmeurs qui s’ennuient et qui en ont marre de jouer au solitaire jouent souvent à un jeu appelé « Des sous dans l’anneau ». Le jeu a pour objectif de déterminer le nombre de pièces d’un cent qui peuvent être placées dans un cercle. Le cercle est tracé dans un grand plan cartésien et son centre est au point $(0, 0)$. Une pièce d’un cent est placée sur chaque point de treillis (c.-à-d. un point dont les deux coordonnées sont des entiers, comme $(1, 1)$, $(1, 2)$, etc.) qui est situé sur le cercle ou à l’intérieur du cercle. Le jeu n’est pas très excitant, mais il s’agit d’un excellent moyen de perdre son temps. Votre but est de déterminer le nombre de pièces d’un cent qu’il faut pour un cercle de rayon donné.

Précisions par rapport aux entrées
Les entrées sont une série d’entiers positifs, un par ligne, représentant chacun le rayon d’un cercle. Vous pouvez supposer que chaque entier sera inférieur ou égal à $25\,000$. Le dernier entier sera un 0. Vous pouvez supposer que le quadrillage du plan cartésien est suffisamment grand pour que deux pièces de monnaie puissent être placées sur des points de treillis adjacents sans se toucher.

Précisions par rapport aux sorties
Pour chaque entrée, vous devez sortir, sur une ligne, le nombre de pièces d’un cent qu’il faut pour ce cercle. Il n’est pas nécessaire de sortir 0 pour la dernière entrée 0. Vous pouvez supposer que le nombre possible de pièces de monnaie est inférieur à 2 milliards (soit 2 mille millions, ce qui équivaut à 20 millions de dollars ; les programmeurs sont riches).

Exemple d’entrées
```
2
3
4
0
```

Sorties pour l’exemple
```
13
29
49
```

Problème S3 : Dédale

Description du problème
Pour gagner un peu d'argent, vous avez décidé de participer à une expérience scientifique. On vous donne à manger des pizzas, encore d'autres pizzas, puis on vous envoie en ville où vous devez trouver votre chemin sur un scooter alimenté à la pizza. La ville compte un grand nombre de carrefours qui sont bien contrôlés. Certains carrefours ont un accès interdit ; certains carrefours vous permettent seulement de continuer en direction est/ouest ; d'autres carrefours vous permettent seulement de continuer en direction nord/sud ; les autres vous permettent de continuer dans n'importe quelle direction, soit nord, sud, est ou ouest.

Heureusement que les scientifiques vous ont remis un plan de la ville (sur le dessous d'une boîte de pizza), avec des symboles sur le plan qui indiquent comment vous pouvez circuler. Il y a 4 symboles différents :

- Le symbole + indique que vous pouvez continuer dans n'importe quelle direction (nord/sud/est/ouest) à partir de cet endroit.
- Le symbole − indique que vous pouvez seulement continuer en direction est ou ouest à partir de cet endroit.
- le symbole | indique que vous pouvez seulement continuer en direction nord ou sud à partir de cet endroit.
- Le symbole $\star$ indique que vous ne pouvez pas occuper cet endroit.

Votre tâche est de terminer le nombre de carrefours que vous devez traverser, y compris les carrefours de départ et d'arrivée, pour aller du coin nord-ouest de la ville jusqu'au coin sud-est de la ville.

Précisions par rapport aux entrées
La première entrée est un entier t ($1 \leq t \leq 10$), sur une ligne, qui indique le nombre de scénarios d'essai contenus dans le fichier. Ensuite, chaque scénario d'essai commence par un entier r, sur une ligne, suivi d'un entier c sur la ligne suivante ($1 \leq r, c \leq 20$). Les r lignes suivantes contiennent chacune c caractères, chaque caractère étant +, $\star$, − ou |. Voux pouvez supposer que le coin nord-ouest de la ville peut être occupé (c.-à-d. qu'il ne sera pas indiqué par un $\star$).

Précisions par rapport aux sorties
La sortie comptera t lignes contenant chacune un entier. L'entier de la ligne i ($1 \leq i \leq t$) indique le nombre de carrefours que vous devez traverser, y compris les carrefours de départ et d'arrivée, pour aller du coin nord-ouest de la ville jusqu'au coin sud-est de la ville. S'il n'y a aucune façon de se rendre du coin nord-ouest jusqu'au coin sud-est, sortez -1 pour ce scénario d'essai.

Exemple d'entrées

```
3
2
2
-|
*+
3
5
+||*+
+++|+
**--+
2
3
+*+
+*+
```

Sorties pour l'exemple

```
3
7
-1
```

Problème S4 : Vingt-quatre

Description du problème

Le Vingt-quatre est un jeu de cartes populaire, pour quatre personnes. Chaque joueur reçoit un jeu de cartes placées faces vers le bas. À chaque tour, chacun des quatre joueurs retourne la carte placée au haut de son jeu, de façon que tous puissent la voir. Chacun tente ensuite de former une expression arithmétique qui a une valeur de 24 en utilisant les valeurs de ces quatre cartes (l'as vaut 1, le valet (J) vaut 11, la dame (Q) vaut 12 et le roi (K) vaut 12). Voici, par exemple, une expression possible pour les quatre cartes illustrées ci-contre :

```
((A * K)- J) * Q
((1*13)-11) * 12
```

Le premier joueur qui obtient une telle expression gagne le tour et ajoute les quatre cartes au bas de son jeu.

Une expression arithmétique valide doit utiliser les quatre cartes, tout en combinant leur valeur en utilisant l'addition, la soustraction, la multiplication ou la division. Les parenthèses peuvent être utilisées pour indiquer la priorité des opérations. Il est interdit de juxtaposer deux cartes pour former un nombre à plusieurs chiffres (p. ex., on ne peut pas juxtaposer les cartes 2 et 4 pour former le nombre 24). Les quotients non entiers d'une division ne sont pas permis, même pas comme réponse intermédiaire (d'une sous-expression de l'expression complète).

Il se peut que les élèves prennent beaucoup de temps pour déterminer une expression ayant une valeur de 24. De fait, il arrivera qu'aucune telle expression n'existe. Étant donné quatre cartes, votre tâche est de déterminer une expression dont la valeur est le plus grand nombre possible inférieur ou égal à 24.

Précisions par rapport aux entrées

La première ligne contient un entier N ($1 \leq N \leq 5$) qui indique le nombre de mains de cartes qui suivent. Chaque main est composée de quatre lignes. Chacune de ces quatre lignes contient un entier C ($1 \leq C \leq 13$) indiquant la valeur d'une carte.

Précisions par rapport aux sorties

Pour chaque main, sortez un entier n sur une ligne. Cet entier n ($n \leq 24$) doit représenter la plus grande valeur possible de toutes les expressions arithmétiques possibles obtenues à partir des quatre cartes de la main.

Exemple d’entrées

```
3
3
3
3
3
1
1
1
1
12
5
13
1
```

Sorties pour l’exemple

```
24
4
21
```

Problème J5 : Réaction

Description du problème
Les deux plus grands physiciens nucléaires canadiens, Patrick et Roland, viennent de compléter la construction du premier réacteur de fission nucléaire au monde. Ils doivent maintenant s'asseoir et faire fonctionner le réacteur toute la journée, chaque jour. Naturellement, après quelque temps ils se sont ennuyés. Pour se distraire, ils ont appris à contrôler les réactions individuelles à l'intérieur du réacteur et ils ont inventé un jeu appelé Réaction.

Dans le jeu Réaction, un certain nombre de particules sont placées dans le réacteur au départ. Les joueurs jouent à tour de rôle, mais Patrick joue toujours le premier. Lorsque c'est à son tour de jouer, un joueur doit choisir un nombre des particules qui restent de manière à former une des réactions possibles. Ces particules sont alors détruites. Éventuellement, il reste tellement peu de particules qu'il devient impossible de former une autre réaction nucléaire ; le joueur dont c'est le tour et qui ne peut jouer perd alors la partie.

Dans notre univers, vous pouvez supposer qu'il n'y a que quatre sortes de particules, soit A, B, C et D. Chaque réaction correspond à une liste de particules qui peuvent être détruites dans un même tour. Voici les cinq réations possibles :

1. AABDD
2. ABCD
3. CCD
4. BBB
5. AD

Par exemple, la première réaction, soit « AABDD », indique qu'il est possible de détruire deux particules A, une particule B et deux particules D dans un même tour.

Il ressort que peu importe le nombre et la sorte de particules qui sont déposées dans le réacteur, exactement l'un des deux joueurs peut avoir une *stratégie gagnante parfaite*. On dit que le *joueur X a une stratégie gagnante parfaite* si, peu importe les réactions choisies par son adversaire, X peut toujours gagner en choisissant ses réactions avec soin. Par exemple, si on a placé une particule A, cinq particules B et trois particules D dans le réacteur, alors Roland a une stratégie gagnante parfaite, soit : « Si Patrick choisit la réaction BBB au départ, Roland choisit la réaction AD en réponse ; si Patrick choisit la réaction AD au départ, Roland choisit la réaction BBB en réponse. » (Cette stratégie est gagnante parce que dans les deux cas, Patrick ne peut jouer à son second tour, puisqu'il ne reste pas assez de particules pour former une des cinq réactions possibles.)

Étant donné un nombre de particules de chaque sorte qui sont déposées dans le réacteur, pouvez-vous déterminer lequel des deux a une stratégie gagnante parfaite ?

Précisions par rapport aux données
La première ligne d'entrée contient le nombre n ($1 \leq n < 100$), soit le nombre de scénarios d'essais. Chaque scénario d'essai est formé de 4 entiers séparés par une espace, tous sur une ligne ; ils représentent le nombre initial respectif de particules A, B, C et D qui sont déposées dans le réacteur. Vous pouvez supposer que chaque nombre initial de particules est un entier de 0 à 30.

Précisions par rapport aux sorties
Pour chaque scénario d'essai, imprimez le nom du joueur qui a une stratégie gagnante parfaite, soit Patrick ou Roland.

Exemple d'entrées
```
6
0 2 0 2
1 3 1 3
1 5 0 3
3 3 3 3
8 8 6 7
8 8 8 8
```

Sorties pour l'exemple
```
Roland
Patrick
Roland
Roland
Roland
Patrick
```

Explication partielle des sorties pour l'exemple
Dans le premier scénario, Roland gagne, puisque Patrick ne peut former *aucune* réaction. (La stratégie de Roland, c'est de ne rien faire.)

Dans le deuxième scénario, Patrick a une stratégie gagnante, soit « former la réaction ABCD », qui a pour effet de faire perdre Roland à son premier tour.

L'issue du troisième scénario est expliquée dans l'énoncé du problème.