

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

*Concours
canadien
d'informatique
2016*

*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. Le concours est conçu pour permettre à la plupart des participants d'obtenir au moins une partie des points disponibles pour chaque problème. Lisez bien et tentez d'obtenir une solution pour chaque problème.
6. Vous devez faire votre propre travail. Les tricheurs seront punis sévèrement.
7. 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.
8. 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.
9. Vous devez vous limiter aux applications de programmation ordinaires (éditeurs, compilateurs, débogueurs). Toutes les autres applications sont **interdites**. Leur utilisation entraînera une disqualification.
10. 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.

11. Votre programme sera exécuté avec des jeux d'essais différents de ceux qui figurent comme exemples. Assurez-vous de vérifier votre programme au moyen d'autres jeux 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.
12. Dans ce concours de niveau supérieur, vous disposerez d'au plus 5 secondes d'exécution par jeu 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 jeu d'essai dont l'exécution prend plus de 5 secondes.
13. Si vous vous placez parmi les 20 premiers participants et participantes dans le concours du niveau supérieur, vous serez invité à participer à l'Olympiade canadienne d'informatique qui aura lieu à l'Université de Waterloo au mois de mai 2016. L'équipe qui représentera le Canada aux Olympiades internationales d'informatique (IOI 2016) en Russie sera choisie parmi les meilleurs candidats de l'Olympiade canadienne. Notez que vous devez connaître C, C++ ou Pascal si vous êtes invité à l'Olympiade canadienne. Mais d'abord, vous devez réussir le concours d'aujourd'hui !
14. 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 : Ragammena

Description du problème

On forme un *anagramme* d'une chaîne en changeant l'ordre des lettres de la chaîne. Par exemple, les anagrammes de *aab* sont *aab*, *aba* et *baa*.

Un *anagramme joker* d'une chaîne est un anagramme de la chaîne dans laquelle certaines des lettres ont peut-être été remplacées par un astérisque (*). Par exemple, deux anagrammes jokers possibles de *aab* sont **ab* et **b**.

Étant donné deux chaînes, déterminer si la deuxième chaîne est un anagramme joker de la première chaîne.

Précisions par rapport aux entrées

Chacune des deux lignes d'entrée contiennent N ($1 \leq N \leq 100$) caractères. Chaque caractère de la première ligne est une lettre minuscule. Chaque caractère de la deuxième ligne est soit une lettre minuscule, soit un astérisque.

Pour 8 des 15 points disponibles, la deuxième ligne ne contiendra aucun astérisque.

Précisions par rapport aux sorties

La sortie sera le caractère A si la chaîne de la deuxième ligne est un anagramme joker de la deuxième ligne. Autrement, la sortie sera le caractère N.

Exemple d'entrée 1

```
abba  
baaa
```

Sortie pour l'exemple d'entrée 1

```
N
```

Exemple d'entrée 2

```
cccrocks  
socc*rk*
```

Sortie pour l'exemple d'entrée 2

```
A
```

Problème S2 : Tandem

Description du problème

Depuis des temps lointains, les habitants du Dmojistan et ceux de Perlande se font la guerre. Or, ils viennent de signer une armistice. Pour célébrer, ils décident qu'ils se promèneront tous en tandems (bicyclettes pour deux). Chaque pays compte N habitants. Ils seront tous placés en couples de manière que chaque couple soit composé d'un habitant du Dmojistan et d'un habitant de Perlande.

Chaque habitant a sa propre vitesse de pédalage. Dans un couple, c'est toujours la personne qui pédale plus vite qui pédalera le tandem et l'autre personne pourra jouir du paysage. En d'autres mots, si les deux personnes qui forment un couple pédalent à des vitesses a et b , alors la *vitesse de tandem* de ce couple est égale à $\max(a, b)$. La *vitesse totale* est la somme des N *vitesses de tandems* individuelles.

Dans chaque jeu d'essai, on vous demandera de répondre à une des questions suivantes :

- Question 1 : Quelle est la vitesse totale minimale parmi toutes les façons de placer les habitants en couples ?
- Question 2 : Quelle est la vitesse totale maximale parmi toutes les façons de placer les habitants en couples ?

Précisions par rapport aux entrées

La première ligne contient un 1 ou un 2 pour indiquer le numéro de la question à laquelle il faut répondre.

La deuxième ligne contient un entier N ($1 \leq N \leq 100$).

La troisième ligne contient N entiers séparés d'une espace. Ces entiers indiquent les vitesses de pédalage des habitants du Dmojistan.

La quatrième ligne contient N entiers séparés d'une espace. Ces entiers indiquent les vitesses de pédalage des habitants de Perlande.

Chaque vitesse sera un entier de 1 à 1 000 000.

Pour 8 des 15 points disponibles, la question 1 sera posée. Pour 7 des 15 points disponibles, la question 2 sera posée.

Précisions par rapport aux sorties

La sortie contiendra un entier qui représente la vitesse totale maximale ou minimale, selon la question posée.

Exemple d'entrée 1

1

3

5 1 4

6 2 4

Sortie pour l'exemple d'entrée 1

12

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

Il y a une solution optimale unique :

- Appairer l'habitant de vitesse 5 (du Dmojistan) et celui de vitesse 6 (de Perlande).
- Appairer l'habitant de vitesse 1 (du Dmojistan) et celui de vitesse 2 (de Perlande).
- Appairer l'habitant de vitesse 4 (du Dmojistan) et celui de vitesse 4 (de Perlande).

Exemple d'entrée 2

2

3

5 1 4

6 2 4

Sortie pour l'exemple d'entrée 2

15

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

Plusieurs solutions optimales sont possibles. En voici une :

- Appairer l'habitant de vitesse 5 (du Dmojistan) et celui de vitesse 2 (de Perlande).
- Appairer l'habitant de vitesse 1 (du Dmojistan) et celui de vitesse 6 (de Perlande).
- Appairer l'habitant de vitesse 4 (du Dmojistan) et celui de vitesse 4 (de Perlande).

Exemple d'entrée 3

2

5

202 177 189 589 102

17 78 1 496 540

Sortie pour l'exemple d'entrée 3

2016

Explication de la sortie pour l'exemple d'entrée 3

Plusieurs solutions optimales sont possibles. En voici une :

- Appairer l'habitant de vitesse 202 (du Dmojistan) et celui de vitesse 1 (de Perlande).
- Appairer l'habitant de vitesse 177 (du Dmojistan) et celui de vitesse 540 (de Perlande).
- Appairer l'habitant de vitesse 189 (du Dmojistan) et celui de vitesse 17 (de Perlande).
- Appairer l'habitant de vitesse 589 (du Dmojistan) et celui de vitesse 78 (de Perlande).
- Appairer l'habitant de vitesse 102 (du Dmojistan) et celui de vitesse 496 (de Perlande).

La vitesse totale est égale à $202 + 540 + 189 + 589 + 496$, ou 2016.

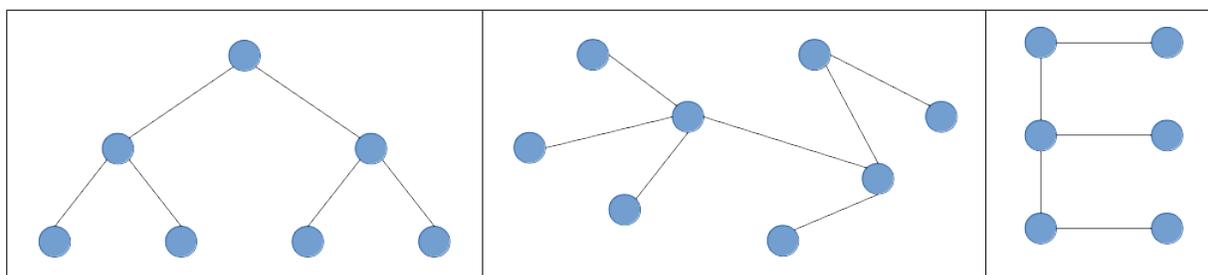
Problème S3 : Critique phonoménale

Description du problème

Jo est blogueuse et se spécialise dans la critique de restaurants. Aujourd'hui, elle veut visiter tous les restaurants vietnamiens de la ville de Waterloo qui offrent de la soupe Pho.

Il existe N restaurants dans la ville de Waterloo, numérotés de 0 à $N - 1$. Or, seulement M de ces restaurants offrent de la soupe Pho. Jo peut choisir de commencer à n'importe quel restaurant. Il y a $N - 1$ rues dans Waterloo, chaque rue reliant exactement deux restaurants. Il est possible de se rendre à tous les restaurants à partir de n'importe quel restaurant en empruntant ces rues. Jo met exactement 1 minute pour parcourir la longueur de n'importe quelle rue.

En informatique, un réseau de rues ayant une telle structure est appelé un *arbre*. Voici trois exemples d'arbres :



La propriété suivante est vraie pour *tous* les arbres : entre n'importe quels deux points dans un arbre, il existe exactement un chemin qui n'emprunte aucune rue plus d'une fois.

Quel est le temps minimal que Jo doit mettre sur la route pour visiter tous les restaurants qui servent de la soupe Pho ?

Précisions par rapport aux entrées

La première ligne d'entrée contient 2 entiers, N et M ($2 \leq M \leq N \leq 100\,000$).

La deuxième ligne d'entrée contient M entiers distincts qui indiquent les restaurants qui servent de la soupe Pho.

Les $N - 1$ lignes suivantes contiennent chacune 2 entiers. La $i^{\text{ème}}$ ligne contient a_i et b_i , ($0 \leq a_i, b_i \leq N - 1$), ce qui indique qu'il y a une rue qui relie a_i et b_i .

Pour 3 des 15 points disponibles, on aura $M = 2$ et $N \leq 100$.

Pour 3 autres des 15 points disponibles, on aura $M \leq 3$ et $N \leq 100$.

Pour 3 autres des 15 points disponibles, on aura $M \leq 8$ et $N \leq 100$.

Pour 4 autres des 15 points disponibles, on aura $M \leq N \leq 1000$.

Précisions par rapport aux sorties

La sortie sera un entier sur une ligne. Il représente le nombre minimal de minutes que Jo passera sur la route pour visiter les restaurants qui servent de la soupe Pho.

Exemple d'entrée 1

8 2
5 2
0 1
0 2
2 3
4 3
6 1
1 5
7 3

Sortie pour l'exemple d'entrée 1

3

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

Le chemin de 5 à 2 va comme suit : $5 \rightarrow 1 \rightarrow 0 \rightarrow 2$. Il utilise 3 rues.

Exemple d'entrée 2

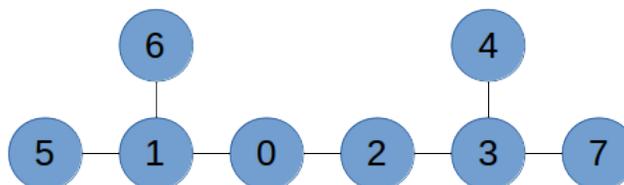
8 5
0 6 4 3 7
0 1
0 2
2 3
4 3
6 1
1 5
7 3

Sortie pour l'exemple d'entrée 2

7

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

Si Jo commence au restaurant 6, elle peut faire un trajet qui n'utilise que 7 rues. Voici un trajet possible : $6 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 3 \rightarrow 4$. On remarque que Jo n'a pas besoin de visiter le restaurant 5, puisqu'il ne sert pas de soupe Pho. La figure suivante représente le réseau de rues :



Problème S4 : Combinaisons de boules de riz

Description du problème

Alphonse a N boules de riz de diverses grandeurs alignées devant lui. Il veut former la plus grande boule de riz possible afin de l'offrir à son ami. Alphonse peut accomplir les opérations suivantes :

- Si deux boules de riz **adjacentes** sont de même grandeur, il peut les combiner pour former une boule plus grande. La grandeur de cette nouvelle boule est la somme des grandeurs des boules utilisées pour la former. La nouvelle boule occupe la position occupée par les deux boules utilisées pour la former.
- Si deux boules de riz sont de même grandeur et qu'il y a **exactement une boule de riz entre elles**, Alphonse peut combiner les trois boules de riz pour former une boule plus grande. (Il n'est pas nécessaire que la boule du milieu ait la même grandeur que les deux autres boules.) La grandeur de la nouvelle boule est la somme des grandeurs des trois boules utilisées pour la former. La nouvelle boule occupe la position occupée par les trois boules utilisées pour la former.

Alphonse peut accomplir chaque opération aussi souvent qu'il veut.

Déterminer la grandeur de la plus grande boule de riz dans la ligne après 0 ou plusieurs combinaisons.

Précisions par rapport aux entrées

La première ligne contient un entier, N ($1 \leq N \leq 400$).

La ligne suivante contient N entiers séparés d'une espace, représentant les grandeurs des boules de riz, de gauche à droite. Chaque entier est supérieur ou égal à 1 et inférieur ou égal à 1 000 000.

- Pour 1 des 15 points disponibles, on aura $N = 4$.
- Pour 2 autres des 15 points disponibles, on aura $N \leq 10$.
- Pour 5 autres des 15 points disponibles, on aura $N \leq 50$.

Précisions par rapport aux sorties

La sortie sera un entier qui représente la grandeur de la plus grande boule de riz qu'Alphonse peut former.

Exemple d'entrée 1

7

47 12 12 3 9 9 3

Sortie pour l'exemple d'entrée 1

48

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

Voici une façon de s'y prendre pour former une boule de grandeur 48 : on combine les deux boules de grandeur 12 pour former une boule de grandeur 24 et les deux boules de grandeur 9 pour former une boule de grandeur 18 (on a alors les boules 47 24 3 18 3). On combine ensuite les boules de grandeurs 3, 18 et 3 pour former une boule de grandeur 24 (on a alors les boules 47 24 24). On combine ensuite les boules de grandeurs 24 pour former une boule de grandeur 48 (on a alors les boules 47 48).

Exemple d'entrée 2

4

1 2 3 1

Sortie pour l'exemple d'entrée 2

3

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

Aucune combinaison n'est possible. La plus grande boule de riz est la plus grande boule dans la ligne, soit la boule 3.

Problème S5 : Cercle de la vie

Description du problème

Vous connaissez peut-être le *Jeu de la vie*, du mathématicien John Conway. Il s'agit d'un ensemble de règles portant sur un quadrillage de manière à produire un ensemble de configurations complexes. Dans ce problème, on traite d'une version simplifiée du jeu.

On considère une bande unidimensionnelle de forme circulaire contenant N cellules. Les cellules sont numérotées de 1 à N dans l'ordre habituel, c'est-à-dire que les cellules 1 et 2 sont adjacentes, les cellules 2 et 3 sont adjacentes et ainsi de suite, jusqu'à la cellule $N - 1$, qui est adjacente à la cellule N . Puisque la bande est circulaire, la cellule 1 est adjacente à la cellule N .

Chaque cellule est soit vivante (que l'on représente par un 1), soit morte (que l'on représente par un 0). L'état des cellules change selon les générations. Si **exactement** un des voisins d'une cellule est vivant dans la génération actuelle, alors la cellule sera vivante dans la génération suivante. Autrement, la cellule sera morte dans la génération suivante.

Étant donné l'état actuel d'une bande, déterminer son état après T générations.

Précisions par rapport aux entrées

La première ligne contient deux entiers, N et T , séparés d'une espace, avec ($3 \leq N \leq 100\,000$ et $1 \leq T \leq 10^{15}$). La deuxième ligne contient une chaîne d'exactly N caractères représentant l'état initial des N cellules. Chaque caractère de la chaîne est un 0 ou un 1. L'état initial de la $i^{\text{ème}}$ cellule est donné par le $i^{\text{ème}}$ caractère de la chaîne. Le caractère 1 indique que la cellule est vivante et le caractère 0 indique que la cellule est morte.

Pour 1 des 15 points disponibles, on aura $N \leq 15$ et $T \leq 15$.

Pour 6 autres des 15 points disponibles, on aura $N \leq 15$.

Pour 4 autres des 15 points disponibles, on aura $N \leq 4000$ et $T \leq 10\,000\,000$.

Pour obtenir le maximum de points, une solution devra pouvoir traiter des entiers de 64 bits. Par exemple,

- en C/C++, il faudrait utiliser une chaîne de type `long long` ;
- en Java, il faudrait utiliser une chaîne de type `long` ;
- en Pascal, il faudrait utiliser une chaîne de type `int64`.

Précisions par rapport aux sorties

La sortie sera une ligne contenant une chaîne de N caractères représentant l'état final des cellules dans le même format et le même ordre que pour l'entrée.

Exemple d'entrée 1

```
7 1
0000001
```

Sortie pour l'exemple d'entrée 1

1000010

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

La cellule 1 et la cellule $N - 1$ sont adjacentes à la cellule N . Elles sont donc vivantes après une génération.

Exemple d'entrée 2

5 3
01011

Sortie pour l'exemple d'entrée 2

10100

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

Après une génération, la chaîne devient 00011.

Après deux générations, la chaîne devient 10111.