



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

*Concours
canadien
d'informatique
2012*

*Niveau
intermédiaire*

Commanditaire :



Concours canadien d'informatique — niveau intermédiaire

Directives à l'intention des participantes et des participants

1. Vous pouvez participer à un concours seulement. Pour participer au concours de niveau supérieur, 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 **intermédiaire**.
3. Vous avez trois (3) heures pour accomplir le travail.
4. Vous pouvez prendre pour acquis que :
 - toutes les entrées se font par le biais du clavier ;
 - 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.* 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 fichier qui sont propres à chaque problème : par exemple, j1.pas ou j1.c ou j1.java (ou tout autre suffixe de fichier approprié) pour le problème J1. Ceci facilitera la tâche de l'évaluateur.
10. Votre programme sera exécuté avec des fichiers d'essais différents de ceux qui figurent comme exemples. Assurez-vous de vérifier votre programme au moyen d'autres fichiers d'essais. Pour certains problèmes, 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 du niveau intermédiaire 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'Ontario
 - Toronto métropolitain
 - Le centre et l'ouest de l'Ontario
 - Le Québec et les provinces de l'Atlantique
12. 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 J1 : Vitesse au volant

Description du problème

Dans plusieurs villes, on place des signaux, munis d'un radar, qui annoncent la vitesse des véhicules dans le but de les faire ralentir.

Vous devez préparer des messages pour un tel signal. Les messages s'adresseront aux chauffeurs selon leur vitesse, tout en tenant compte du tableau suivant :

Nombre de km/h au-dessus de la limite	Amende
1 à 20	100 \$
21 à 30	270 \$
31 ou plus	500 \$

Précisions par rapport aux entrées

L'utilisateur sera sollicité d'entrer deux nombres entiers. D'abord, on lui demandera d'entrer la limite de vitesse permise. Ensuite, on lui demandera d'entrer la vitesse de la voiture enregistrée par le radar.

Précisions par rapport aux sorties

Si la voiture n'excède pas la limite permise, la sortie doit être :

Bravo ! Vous respectez la limite permise.

Si la voiture excède la limite permise, la sortie doit être :

Exces de vitesse, amende de F \$.

F étant l'amende à payer selon le tableau précédent.

Exemple 1 (les entrées en texte romain, les sorties en italique)

Entrer la limite permise : *40*

Entrer la vitesse de la voiture selon le radar : *39*

Bravo ! Vous respectez la limite permise.

Exemple 2

Entrer la limite permise : *100*

Entrer la vitesse de la voiture selon le radar : *131*

Exces de vitesse, amende de 500 \$.

Exemple 3

Entrer la limite permise : *100*

Entrer la vitesse de la voiture selon le radar : *120*

Exces de vitesse, amende de 100 \$.

Problème J2 : Histoire de pêche !

Description du problème

Un détecteur de poissons est un dispositif qui permet aux pêcheurs de trouver des poissons dans un lac ou une rivière. Si le détecteur trouve un poisson, il fait sonner une alarme. Le détecteur fait appel à des prélèvements de profondeur pour déterminer si l'alarme doit sonner. Dans notre cas, le détecteur de poissons décidera qu'un poisson passe en dessous de nous si :

- il y a quatre prélèvements de profondeur consécutifs qui forment une suite strictement croissante (p. ex., 3 4 7 9) (on dira « Poisson qui monte »), ou
- il y a quatre prélèvements de profondeur consécutifs qui forment une suite strictement décroissante (p. ex., 9 6 5 2) (on dira « Poisson qui descend »), ou
- il y a quatre prélèvements de profondeur consécutifs identiques (on dira « Poisson à profondeur constante »).

Tous les autres prélèvements de profondeur seront considérés comme des bruits aléatoires (on dira « Aucun poisson »).

Votre programme doit lire une suite de prélèvements et déterminer si l'alarme doit sonner.

Précisions par rapport aux entrées

L'entrée est constituée de quatre lignes, chaque ligne contenant un entier strictement positif qui représente un prélèvement.

Précisions par rapport aux sorties

Il y a quatre possibilités. Si les prélèvements sont quatre entiers croissants, la sortie doit être `Poisson qui monte`. Si les prélèvements sont quatre entiers décroissants, la sortie doit être `Poisson qui descend`. Si les prélèvements sont quatre entiers identiques, la sortie doit être `Poisson a profondeur constante`. Dans tout autre cas, la sortie doit être `Aucun poisson`.

Exemple d'entrée 1

```
30
10
20
20
```

Sortie pour l'exemple d'entrée 1

```
Aucun poisson
```

Exemple d'entrée 2

```
1
10
12
13
```

Sortie pour l'exemple d'entrée 2

```
Poisson qui monte
```

Problème J3 : Icône à l'échelle

Description du problème

On vous demande de regarder une icône qui paraît sur l'écran d'un téléphone intelligent et de l'agrandir à l'échelle pour qu'elle paraisse plus grand sur un écran régulier d'ordinateur.

L'icône sera encodée au moyen des caractères (x et *) dans un tableau 3×3 comme suit :

```
*x*
  xx
*  *
```

Vous devez écrire un programme qui reçoit un facteur d'agrandissement et qui produit l'icône agrandie à l'échelle. Un facteur de d'agrandissement k indique que chaque caractère est remplacé par un tableau $k \times k$ rempli de ce caractère.

Précisions par rapport aux entrées

L'entrée sera un entier strictement positif k tel que $k < 25$.

Précisions par rapport aux sorties

La sortie sera formée de $3k$ lignes. Chaque caractère d'une ligne du tableau de l'icône initiale paraît k fois de suite et la ligne qui en résulte paraît k fois de suite.

Exemple d'entrée

3

Sortie pour l'exemple d'entrée

```
***xxx***
***xxx***
***xxx***
  xxxxxx
  xxxxxx
  xxxxxx
***    ***
***    ***
***    ***
```

Problème J4 : Secrets du Big Bang

Description du problème

Gregory et Charles sont deux scientifiques passionnés de la théorie du BIG BANG. Afin d'échanger sur leurs idées en secret, ils ont conçu un code qui encode les mots en MAJUSCULES en remplaçant chaque lettre par une lettre plus loin dans l'alphabet.

Le déplacement d'une lettre de S positions signifie que la lettre est remplacée par la lettre qui est à S positions plus loin dans l'alphabet. Par exemple, le déplacement de la lettre B de S positions, lorsque $S = 3$, donne la lettre E. Lorsque le déplacement nous fait aller au-delà de la lettre Z, la dernière lettre de l'alphabet, on procède en boucle, en considérant A comme la lettre qui suit la lettre Z. Par exemple, le déplacement de la lettre Z de S positions, lorsque $S = 2$, donne la lettre B.

Le code dépend du paramètre K ainsi que de la position de chaque lettre dans le mot. Si la lettre est en position P dans le mot, la valeur du déplacement est égale à $D = 3P + K$.

Par exemple, voici comment on encode le mot ZOOM lorsque $K = 3$. Pour la 1^{re} lettre, Z, on a $D = 3 \times 1 + 3$, ou $D = 6$, ce qui nous mène à la lettre F. Pour la 2^e lettre, O, on a $D = 3 \times 2 + 3$, ou $D = 9$, ce qui nous mène à la lettre X. Les deux dernières lettres deviennent A et B. Donc, Gregory envoie le message suivant à Charles : FXAB

Vous devez écrire un programme qui **décode** les messages envoyés par Gregory.

Précisions par rapport aux entrées

L'entrée est constituée de deux lignes. La première ligne contient un entier strictement positif K ($K < 10$) qui est la valeur du paramètre utilisé dans la formule du déplacement. La deuxième ligne est un mot, soit une séquence de 20 lettres majuscules ou moins.

Précisions par rapport aux sorties

La sortie est le mot décodé écrit en majuscules.

Exemple d'entrée 1

3
FXAB

Sortie pour l'exemple d'entrée 1

ZOOM

Exemple d'entrée 2

5
JTUSUKG

Sortie pour l'exemple d'entrée 2

BIGBANG

Problème J5 : Jeu de monnaie

Description du problème

Lorsqu'elle s'ennuie, Anne Coderre aime jouer au jeu suivant sur une table à l'aide de pièces de monnaie. Elle prend un ensemble de pièces distinctes de monnaie qu'elle place en ligne droite. Par exemple, supposons qu'elle a une pièce U de 1 ¢ (ou 0,01 \$), une pièce C de 5 ¢ (ou 0,05 \$) et une pièce D de 10 ¢ (ou 0,10 \$). Elle les place en ligne droite de façon arbitraire (par exemple, D C U) et elle les déplace de manière à les replacer en ordre strictement croissant selon leur valeur, par exemple, U C D (c.-à-d. 0,01 \$, 0,05 \$, 0,10 \$). Voici les règlements du jeu :

- L'alignement initial des pièces de monnaie définit toutes les positions dans lesquelles on peut placer les pièces. Donc, on ne peut pas ajouter des positions pendant le jeu et même si une position n'est pas occupée à un moment particulier du jeu, la position continue d'exister.
- Le jeu consiste en une séquence de déplacements et à chaque déplacement, une pièce est déplacée à une position *adjacente*.
- Les pièces de monnaie peuvent être empilées les unes sur les autres et lors d'un déplacement, Anne prend toujours une pièce sur le dessus d'une pile pour la placer sur le dessus d'une pile adjacente.
- Lorsqu'elle place une pièce dans une pile, Anne ne peut pas la placer sur une pièce de valeur inférieure.

Pour simplifier, on donnera aux pièces de monnaie des valeurs qui sont des entiers consécutifs (p. ex., la pièce de 1 ¢ vaut 1, la pièce de 5 ¢ vaut 2 et la pièce de 10 ¢ vaut 3). Dans l'exemple précédent, Anne réussirait en 20 déplacements. Le tableau suivant indique les déplacements (XY signifie que la pièce X est par-dessus la pièce Y) :

Dépl.	Position 1	Position 2	Position 3
Départ	3	2	1
1	3	12	
2	13	2	
3	13		2
4	3	1	2
5	3		12
6		3	12
7		13	2
8	1	3	2
9	1	23	
10		123	

Dépl.	Position 1	Position 2	Position 3
11		23	1
12	2	3	1
13	2	13	
14	12	3	
15	12		3
16	2	1	3
17	2		13
18		2	13
19		12	3
20	1	2	3

À noter que pour certains alignements initiaux, il est impossible de réussir à replacer les pièces en ordre strictement croissant.

Précisions par rapport aux entrées

L'entrée est constituée d'un certain nombre de jeux d'essais. Un jeu d'essais contient deux lignes. La première ligne contient un entier n strictement positif ($n < 5$), ce qui correspond au nombre de pièces de monnaie. On peut supposer que les pièces auront pour valeurs $1, 2, 3, \dots, n$. La deuxième ligne contient les entiers de 1 à n placés dans un ordre quelconque, ce qui constitue la position de départ. Dans l'exemple précédent, l'entrée serait :

```
3
3 2 1
```

La fin des jeux d'essais est indiquée par un 0 seul sur une ligne.

Précisions par rapport aux sorties

Pour chaque jeu d'essais, la sortie sera une ligne qui indique soit le *nombre minimal* de déplacements qu'Anne peut faire pour réussir à placer les pièces en ordre ou, si c'est impossible de réussir, la ligne indiquera le mot IMPOSSIBLE.

Exemple d'entrée

```
3
3 2 1
2
2 1
0
```

Sortie pour l'exemple d'entrée

```
20
IMPOSSIBLE
```