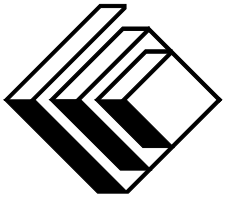

Le Centre d'éducation en mathématiques et en informatique

Concours canadien d'informatique 2007

Niveau intermédiaire

Commanditaire :

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 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.*, 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, `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'essai différents de ceux qui figurent comme exemples. Assurez-vous de vérifier votre programme au moyen d'autres fichiers d'essai.
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'Ontatio
   – 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, pour voir comment on pouvait résoudre les problèmes et pour connaître le nom des gagnants. Voici l'adresse :

    `www.cemc.uwaterloo.ca/ccc`

Problème J1 : Qui est au milieu ?

Description du problème
Dans l'histoire *Boucle d'or et les trois ours*, chaque ours a un bol de soupe devant lui et chacun est assis sur sa chaise préférée. Ce que l'histoire ne nous dit pas, c'est que Boucle d'or a bougé les bols et ceux-ci ne se trouvent plus à leur place appropriée. Or, il est possible d'assortir les bols selon leur poids. Le bol le plus léger appartient à Bébé ours, le bol de poids moyen appartient à Maman ours et le bol le plus lourd appartient à Papa ours. Écrivez un programme qui lira trois poids et qui imprimera le poids du bol de Maman ours. Vous pouvez prendre pour acquis que tous les poids sont des entiers strictement positifs inférieurs à 100.

Exemple d'entrée
```
10
5
8
```

Sortie pour l'exemple
```
8
```

Problème J2 : Je parle MSGTXT

Description du problème
La messagerie texte est populaire chez les adolescents. Les messages peuvent paraître bizarres, parce que des abréviations et des symboles sont utilisés pour réduire le nombre de lettres qu'il faut taper.

Par exemple, « OQP » signifie « occupé » et « :-) » est une binette. Elle représente une figure joyeuse (lorsqu'on la regarde de côté) et indique un sourire. Pour certains adultes, ces textes sont plutôt mystérieux.

Écrivez un programme qui acceptera une abréviation comme entrée et qui produira, comme sortie, la traduction selon le tableau suivant :

Abréviation	Traduction
A+	A plus tard
:-)	Je suis content
:-(	Je suis triste
;-)	Clin d'oeil
(˜.˜)	Je m'endors
MDR	Mort de rire
CCI	Concours canadien d'informatique
PTET	Peut-être
OQP	Occupé
PKOI	Pourquoi
ALP	A la prochaine

Précisions par rapport aux entrées
Le programme doit solliciter l'utilisateur à entrer un texte une ligne à la fois. Lorsque l'abréviation « ALP » est entrée, le programme est terminé et cesse de solliciter des entrées. Les utilisateurs peuvent entrer des abréviations du tableau, ou ils peuvent entrer un texte différent. Toutes les entrées seront formées de symboles ou de lettres majuscules. Elles ne comporteront aucun espace et aucun guillemet.

Précisions par rapport aux sorties
Le programme imprime à l'écran sur la ligne qui suit le texte d'entrée. Si l'entrée est une des abréviations du tableau, la sortie est la traduction ; si l'entrée n'est pas une abréviation du tableau, la sortie est identique à l'entrée. La traduction de la dernière entrée, soit « ALP » , doit être imprimée à l'écran.

Exemple d’une session d’entrées et de sorties (entrées de l’utilisateur en *italique*)

```
Entrez une abréviation> CCI
Concours canadien d’informatique
Entrez une abréviation> :-)
Je suis content
Entrez une abréviation> SQL
SQL
Entrez une abréviation> ALP
A la prochaine
```

Problème J3 : Marché conclu

Description du problème
Le réseau américain de télévision NBC a une émission de jeu qui s'intitule *Deal or No Deal* (Marché conclu ou non) . (On peut jouer à l'adresse : `http://www.nbc.com/Deal_or_No_Deal/game/flash.shtml`)

Dans la version CCI de ce jeu, dix sommes d'argent sont disponibles : 100 $, 500 $, 1000 $, 5000 $, 10 000 $, 25 000 $, 50 000 $, 100 000 $, 500 000 $, 1 000 000 $. Ces sommes sont cachées dans des mallettes imaginaires. Les sommes sont numérotées, dans l'ordre, de 1 à 10 (c'est-à-dire que $1 \rightarrow 100$ $, $2 \rightarrow 500$ $, $3 \rightarrow 1000$ $, ..., $10 \rightarrow 1\,000\,000$ $). Au début du jeu, on choisit une mallette qu'on pourra peut-être garder. Il est interdit de l'ouvrir. Pendant le jeu, on choisit quelques autres mallettes pour en connaître le contenu, ce qui élimine les choix possibles quant au contenu de la mallette choisie au départ.

À un moment donné, on cesse d'ouvrir les mallettes et un « banquier » nous offre de l'argent comptant en échange de la mallette choisie au départ et de son contenu. On nous demande ensuite « Marché conclu ou non ? ».

Écrivez un programme qui nous aide à calculer la moyenne des sommes qui restent dans les mallettes non ouvertes, y compris dans celle qui a été choisie au départ et qui comparera cette moyenne à l'offre du banquier. Si l'offre est plus grande que la moyenne, on devrait l'accepter ; sinon, on ne devrait pas l'accepter.

Précisions par rapport aux entrées
L'utilisateur doit entrer un entier n $(1 \leq n < 10)$ qui indique le nombres de mallettes qui ont été ouvertes jusqu'à ce moment-ci, n entiers de 1 à 10 qui représentent les valeurs dans les mallettes qui ont été ouvertes, suivis de l'offre du banquier. Par exemple : `3 2 5 10 300` indique que les mallettes contenant les sommes de 500 $, 10 000 $ et 1 000 000 $ ont été ouvertes et que le banquier a fait une offre de 300 $. Vous pouvez supposer qu'aucun numéro de mallette ne paraîtra plus d'une fois et que l'offre du banquier sera un entier supérieur à 10.

Précisions par rapport aux sorties
Le programme imprimera « MARCHE CONCLU » ou « NON ».

Exemple d'entreée 1
```
2
3
8
198000
```

Sortie pour l'exemple 1
```
NON
```

Exemple d’entreée 2

```
8
10
9
8
7
6
5
4
3
400
```

Sortie pour l’exemple 2

```
MARCHE CONCLU
```

Problème J4 : Vérificateur d'anagrammes

Description du problème
Une anagramme est un mot ou une expression obtenu à partir de la transposition des lettres d'un autre mot ou d'une autre expression. Par exemple, BARRE est une anagramme de ARBRE. UN VETO CORSE LA FINIRA est une anagramme de REVOLUTION FRANCAISE. Une anagramme peut donc être composée de plusieurs mots. Dans ce cas, il n'est pas nécessaire que les mots de l'anagramme aient les mêmes nombres de lettres que ceux de l'expression initiale. C'est le cas du deuxième exemple. Écrivez un programme qui vérifie si une expression est une anagramme d'une autre expression.

Précisions par rapport aux entrées
Le programme devrait solliciter deux expressions, une par ligne. Vous pouvez prendre pour acquis que l'entrée ne comprend que des lettres majuscules, sans accents, et des espaces.

Précisions par rapport aux sorties
Le programme imprimera une des phrases suivantes, soit « Est une anagramme. » ou « N'est pas une anagramme. ».

Exemple de sollicitation et d'entrée (entrée en *italique*)
```
Entrez la première expression> PABLO PICASSO
Entrez la deuxième expression> PASCAL OBISPO
```

Sortie pour l'exemple
```
Est une anagramme.
```

Problème J5 : On roule

Description du problème
Un camionneur doit voyager de Vancouver à St-Jean Terre-Neuve, sur la route trans-canadienne, soit une distance de 7000 km. Chaque soir, il doit s'arrêter à un motel. On lui a remis une liste des motels qu'il peut choisir, de même que la distance de chaque motel, en km, depuis Vancouver. Voici quelques-unes des distances :

0, 990, 1010, 1970, 2030, 2940, 3060, 3930, 4060, 4970, 5030, 5990, 6010, 7000

Il est possible d'ajouter d'autres motels avant le départ.

Déterminez s'il est possible pour le camionneur de compléter son trajet, sachant que :

1. son employeur insiste pour qu'il roule au moins A km par jour,
2. la loi lui interdit de rouler plus de B km par jour, et
3. chaque soir, il doit choisir un motel qui fait partie de la liste (ceux de la liste précédente ou ceux qui seront ajoutés ci-dessous).

Le camionneur veut connaître ses choix et vous devez écrire un programme qui calculera le nombre de choix à sa disposition.

Par exemple, s'il n'y a aucun ajout à la liste précédente, si $A = 1$ et si $B = 500$, alors il est impossible de compléter le trajet. Le nombre de choix est donc égal à 0. Si $A = 970$ et $B = 1030$, il y a une seule façon de compléter le trajet ; si $A = 970$ et $B = 1040$, il y a quatre façons de compléter le trajet. Si on ajoute un motel à 4960 km, si $A = 970$ et si $B = 1030$, il y a deux façons de compléter le trajet.

Précisions par rapport aux entrées
Les deux premières lignes d'entrée contiennent la distance minimale A et la distance maximale B ($1 \leq A \leq B \leq 7000$). Chaque distance est un entier. La troisième ligne contient un entier N ($0 \leq N \leq 20$), suivi de N lignes donnant chacune la distance m ($0 < m < 7000$), depuis Vancouver, d'un motel acceptable. Il n'y aura pas deux motels situés à une même distance depuis Vancouver.

Précisions par rapport aux sorties
Le programme imprime sur l'écran le nombre de choix offerts au camionneur selon les contraintes.

Exemples d’entrées et de sorties

Exemples d’entrées	Exemples de sorties
1 500 0	0
970 1030 0	1
970 1040 0	4
970 1030 1 4960	2