

Formulaire d'inscription

À photocopier et à renvoyer à l'adresse postale suivante avant le 1^{er} février 2013:

- ☒ Concours Informatique Luxembourgeois
p.a. Centre de Technologie de l'Éducation
29 avenue J.-F. Kennedy
L-1855 Luxembourg

Nom et prénom

Lieu et date de naissance

Adresse privée (rue et numéro, code postal et localité)

Adresse courriel

Téléphone

ATTENTION: vérifiez bien votre adresse courriel, car elle sera le seul moyen de communication entre les organisateurs du CIL et les candidats.

Etablissement scolaire

Classe

Langage(s) de programmation utilisé(s) - cochez ce qui convient: Pascal C(++) Java

L'autorisation d'un des parents ou du représentant légal est indispensable pour l'inscription si le candidat est mineur.

Accord et signature d'un des parents ou du représentant légal (si le candidat est mineur).

Je soussigné(e) _____ m'inscris au Concours Informatique Luxembourgeois 2013 et m'engage à respecter le règlement ainsi que les conditions de participation.

Date et signature du candidat: _____

CENTRE DE
TECHNOLOGIE
DE L'ÉDUCATION
www.cte.lu

22^e CONCOURS
INFORMATIQUE
LUXEMBOURGEOIS
cil.cte.lu

25^e OLYMPIADE
INTERNATIONALE
EN INFORMATIQUE
www.ioi2013.org

22^e
**CONCOURS
INFORMATIQUE
LUXEMBOURGEOIS
2013**

cil.cte.lu

Les quatre gagnants participeront à la
25^e Olympiade Internationale en Informatique
du 6 au 13 juillet 2013
à Brisbane, Australie

Gil Aversy Sabotina

Le Concours Informatique Luxembourgeois (CIL)

Le Concours Informatique Luxembourgeois (CIL) est un concours scolaire de programmation, organisé annuellement par le Centre de Technologie de l'Éducation (CTE). Le concours vise à identifier de jeunes élèves ayant des compétences particulières dans la résolution de problèmes sur ordinateur, dans la conception d'algorithmes et l'implémentation de programmes à l'aide d'un langage de programmation. En 2013, le CIL est organisé déjà pour la 22^e fois. Le concours est ouvert à tous les élèves des lycées publics et privés des enseignements secondaire et secondaire technique, sans limite d'âge. Les langages de programmation admis lors du CIL sont Pascal, C/C++ et Java.



Le CIL 2013 est organisé en quatre étapes:

- l'Épreuve de Sélection Préliminaire,
- l'Épreuve Demi-Finale,
- un certain nombre de Séances de Formations et
- l'Épreuve Finale.

Toutes les informations relatives au concours peuvent être consultées sur le site web cil.cte.lu.

Traditionnellement, les quatre lauréats du CIL sont automatiquement sélectionnés pour faire partie de la délégation luxembourgeoise officielle représentant les couleurs du Grand-Duché de Luxembourg à l'Olympiade Internationale en Informatique (IOI) de la même année et ceci après une formation algorithmique appropriée.

L'Olympiade Internationale en Informatique (IOI)

L'Olympiade Internationale en Informatique (IOI) est une compétition internationale d'algorithmique et de programmation d'ordinateurs pour jeunes élèves. Elle est organisée chaque année dans un autre pays du monde et accueille actuellement les délégations de plus de 80 pays du monde entier. L'IOI est une des sept olympiades scientifiques internationales pour jeunes élèves (les autres concernent les mathématiques, la physique, la biologie, l'astronomie, la chimie et la géographie). Le Grand-Duché de Luxembourg participe régulièrement depuis 1992 aux IOI. *Les langages de programmation officiels des IOI sont uniquement le Pascal et le C/C++.* Les vainqueurs des IOI feront partie des meilleurs programmeurs au monde.



La 25^e édition de l'IOI se déroulera du 6 au 13 juillet 2013 à Brisbane en Australie. Toutes les informations relatives à l'IOI 2013 peuvent être consultées sur le site web www.ioi2013.org.

Calendrier

5 novembre 2012	Lancement officiel du 22 ^e Concours Informatique Luxembourgeois (CIL) 2013
1 ^{er} février 2013	Épreuve de Sélection Préliminaire: date limite des envois (formulaire d'inscription et solutions aux problèmes)
7 mars 2013	Épreuve Demi-Finale
avril/mai 2013	Séances de Formation (dates à déterminer)
16 mai 2013	Épreuve Finale
juin 2013	Formation algorithmique appropriée pour les quatre participants à l'IOI (dates à déterminer)
6-13 juillet 2013	25 ^e Olympiade Internationale en Informatique (IOI) à Brisbane en Australie
15 juillet 2013	Remise des prix du CIL 2013 au Centre de Technologie de l'Éducation

Déroulement du concours

La 22^e édition du Concours Informatique Luxembourgeois (CIL) se déroule en quatre étapes:

Étape I

Épreuve de Sélection Préliminaire

Les candidats doivent résoudre à domicile les problèmes énoncés dans le questionnaire ci-dessous. Le questionnaire se compose de quatre problèmes d'un degré de difficulté varié. Afin de pouvoir participer au concours, il faut avoir résolu au minimum trois problèmes, par exemple les trois premiers (il n'est donc pas nécessaire de résoudre la totalité des problèmes posés, le quatrième étant un «challenge»). Les langages de programmation permis sont Pascal (→ Turbo Pascal, FreePascal, fonctionnalité «console application» de Delphi etc.), C/C++ (→ Turbo C/C++, GNU C/C++ etc.) et Java (NetBeans etc.). Les solutions aux problèmes (sous forme de fichiers source) doivent parvenir aux organisateurs par courrier électronique pour le 1^{er} février 2013 au plus tard à l'adresse électronique suivante:

@ Adresse électronique:

cil@cte.lu

Le formulaire d'inscription dûment rempli et signé doit aussi parvenir aux organisateurs par courrier postal pour le 1^{er} février 2013 au plus tard. Pour les candidats mineurs, l'autorisation d'un des parents ou du représentant légal est indispensable pour l'inscription.

✉ Adresse postale:

Concours Informatique Luxembourgeois
p.a. Centre de Technologie de l'Éducation
29 avenue J.-F. Kennedy
L-1855 Luxembourg

Le questionnaire de l'Épreuve de Sélection Préliminaire ainsi que le formulaire d'inscription sont aussi disponibles sur le site web cil.cte.lu. Les candidats ayant réalisé les meilleurs scores à l'Épreuve de Sélection Préliminaire sont admis à l'Épreuve Demi-Finale.

Étape II

Épreuve Demi-Finale

L'Épreuve Demi-Finale consiste à résoudre individuellement trois problèmes d'un degré de difficulté varié dans un temps déterminé. Les candidats disposent de quatre heures pour concevoir les solutions aux problèmes d'algorithmique posés et pour implémenter les programmes correspondants. L'Épreuve Demi-Finale a lieu le 7 mars 2013 dans une salle informatique d'un lycée. L'Épreuve Demi-Finale permet de sélectionner au plus douze candidats qui sont admis aux Séances de Formation.

Étape III

Séances de Formation

Les Séances de Formation permettent aux candidats sélectionnés d'approfondir leurs connaissances en programmation et de s'approprier des méthodes d'algorithmique. Les Séances de Formation, qui sont au nombre de trois, se déroulent pendant les mois d'avril et de mai 2013. En principe, tous les candidats ayant participé aux séances sont admis à l'Épreuve Finale.

Étape IV

Épreuve Finale

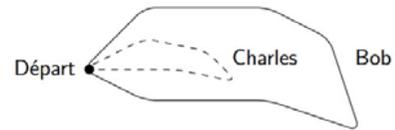
L'Épreuve Finale consiste à résoudre individuellement deux problèmes d'un degré de difficulté varié dans un temps déterminé. Les candidats disposent de quatre heures pour concevoir les solutions aux problèmes d'algorithmique posés et pour implémenter les programmes correspondants. Les problèmes posés requièrent en grande partie la mise en œuvre des méthodes algorithmiques traitées lors des Séances de Formation. L'Épreuve Finale a lieu le 16 mai 2013 dans une salle informatique d'un lycée.

Les quatre meilleurs élèves du concours sont les lauréats du CIL et recevront des prix. Les autres finalistes recevront des prix d'encouragement.

Problème I

Jogging

Bob et Charles ont décidé d'aller courir un peu pour garder la forme. Ils ont chacun leur parcours favori qui part du même point et qui y revient également.



Tâche

Le trajet de Bob fait A km de long tandis que celui de Charles fait B km de long. En supposant que A et B sont des entiers naturels non nuls, écrivez un programme qui calcule le nombre de tours que vont faire Bob et Charles avant de se recroiser au point de départ pour la première fois.

Restrictions

A et B sont des entiers naturels non nuls inférieurs ou égaux à 100.

Entrée et sortie du programme

- Le programme lit d'abord à partir du clavier la distance A du trajet de Bob et puis la distance B du trajet de Charles.
- Ensuite, le programme affiche à la console le nombre de tours que vont faire les deux avant de se recroiser au point de départ pour la première fois. Attention: l'affichage doit être grammaticalement correct (exemple: "1 tour", mais "3 tours").

Exemples d'exécution

Exemple 1

```
Distance du trajet de Bob: 5
Distance du trajet de Charles: 3
Bob va faire 3 tours et Charles va faire 5 tours.
```

Exemple 2

```
Distance du trajet de Bob: 2
Distance du trajet de Charles: 8
Bob va faire 4 tours et Charles va faire 1 tour.
```

Remettez le programme sous le nom JOGGING{.PAS; .C; .CPP; .JAVA}.

Problème II

Jill et Pascale

Jill et Pascale viennent de terminer leurs études secondaires et discutent de ce qu'elles envisagent de faire maintenant. Jill, qui n'a plus envie de faire des études complémentaires, a trouvé un travail à plein temps, où elle gagne un salaire mensuel modeste S_{Jill} . Pascale considère de faire une formation supplémentaire de X années pour ensuite aller travailler en tant que travailleur qualifié pour un salaire mensuel $S_{Pascale}$.

Tâche

Écrivez un programme qui permet de calculer, connaissant le salaire mensuel S_{Jill} de Jill, le salaire mensuel $S_{Pascale}$ de Pascale et la durée de formation X de Pascale, combien de mois Pascale doit travailler pour avoir gagné autant d'argent que Jill, c.-à-d. pour compenser l'avance de Jill.

Restrictions

- $1 \leq S_{Pascale} \leq 10000$ Le salaire mensuel de Pascale.
- $1 \leq S_{Jill} < S_{Pascale}$ Le salaire mensuel de Jill.
- $1 \leq X \leq 10$ La durée de formation de Pascale.

Le salaire mensuel de Pascale, le salaire mensuel de Jill et la durée de formation de Pascale sont des entiers naturels. Attention: les nombres à manipuler risquent devenir grands. On ignore toute anomalie ou irrégularité dans les salaires mensuels (Bonus, 13^e mois, ...).

Entrée et sortie du programme

- Le programme lit d'abord à partir du clavier le salaire mensuel de Jill, puis le salaire mensuel de Pascale et finalement la durée de formation de Pascale (en années).
- Ensuite, le programme affiche à la console le nombre de mois que Pascale doit travailler pour avoir gagné autant d'argent que Jill.

Exemples d'exécution

Exemple 1

```
Salaire de Jill: 1570
Salaire de Pascale: 1884
Durée de la formation de Pascale: 3
Après 180 mois, Pascale a compensé l'avance de Jill
```

Exemple 2

```
Salaire de Jill: 1800
Salaire de Pascale: 3800
Durée de la formation de Pascale: 5
Après 54 mois, Pascale a compensé l'avance de Jill
```

Remettez le programme sous le nom JILL_ET_PASCALE{.PAS; .C; .CPP; .JAVA}.

Problème III

Sudoku

Le but du Sudoku est de remplir une grille avec une série de chiffres tous différents, qui ne se trouvent jamais plus d'une fois sur une même ligne, dans une même colonne ou dans une même sous-grille. Ici, les chiffres sont des entiers de 1 à 9, les sous-grilles étant des carrés de 3×3 . Quelques chiffres sont déjà disposés dans la grille, ce qui autorise une résolution progressive du problème complet.

Tâche

Soit une grille de Sudoku partiellement ou complètement remplie. Le but du programme est de vérifier si la grille est correctement formée. *Il ne s'agit pas de trouver une solution au Sudoku.*

Exemples

5	6	1	4	8	7		9	2
4	9	8	2	1	3		6	7
3	2		5	6	9		4	8
2	5	4	7	9	1	6	8	3
7	8	9	3	2	6	4	1	5
1	3	6	8		4	7	2	9
6	1	3	9	7	2		5	4
9	7	5	6	4	8	2	3	1
8	4	2	1	3	5		7	6

Le Sudoku est correctement formé.

5	6	3	4	8	7	3	9	2
4	9	8	2	1	3	5	6	7
3	2	7	5	6	9	1	4	8
2	5	4	7	9	1	6	8	3
7	8	9	3	2	6	4	1	5
1	3	6	8	5	4	7	2	9
6	1	3	9	7	2	8	5	4
9	7	5	6	4	8	2	3	1
8	4	2	1	3	5	9	7	6

Le Sudoku n'est PAS correctement formé car le chiffre 3 apparaît 2 fois dans la même colonne.

Entrée et sortie du programme

- Le programme lit à partir de la console le nom du fichier contenant la grille du Sudoku formé de 9 lignes de chiffres entiers compris entre 0 et 9 inclus. Le 0 désigne une case vide. Dans une même ligne, deux chiffres sont séparés par un seul espace.
- Le programme doit afficher à la console le message «Le Sudoku est correctement formé» si le Sudoku est correctement formé, c'est-à-dire, si chaque ligne et chaque colonne de la grille ainsi que chaque sous-grille 3×3 ne contient pas plus d'une seule fois un des entiers 1 à 9. Dans le cas contraire, le programme doit afficher «Le Sudoku n'est PAS correctement formé».

Exemple d'exécution 1

Contenu du fichier IN1.TXT:

```
5 6 1 4 8 7 0 9 2
4 9 8 2 1 3 0 6 7
3 2 0 5 6 9 0 4 8
2 5 4 7 9 1 6 8 3
7 8 9 3 2 6 4 1 5
1 3 6 8 0 4 7 2 9
6 1 3 9 7 2 0 5 4
9 7 5 6 4 8 2 3 1
8 4 2 1 3 5 0 7 6
```

Saisie à la console: IN1.TXT

Affichage à la console: Le Sudoku est correctement formé

Exemple d'exécution 2

Contenu du fichier IN2.TXT:

```
5 6 3 4 8 7 3 9 2
4 9 8 2 1 3 5 6 7
3 2 7 5 6 9 1 4 8
2 5 4 7 9 1 6 8 3
7 8 9 3 2 6 4 1 5
1 3 6 8 5 4 7 2 9
6 1 3 9 7 2 8 5 4
9 7 5 6 4 8 2 3 1
8 4 2 1 3 5 9 7 6
```

Saisie à la console: IN2.TXT

Affichage à la console: Le Sudoku n'est PAS correctement formé

Remettez le programme sous le nom SUDOKU{.PAS; .C; .CPP; .JAVA}.

Problème IV - CHALLENGE

SoupAutomat

Le SoupAutomat est un automate cellulaire à une seule dimension¹. Il se compose d'une ligne de 80 cellules, chacune pouvant contenir le symbole espace représenté par un '.' (point) ou un chiffre de '0' à '9'. Le but du programme est de faire passer cette ligne dans l'état suivant en respectant les règles indiquées ci-dessous et ceci un nombre maximal de fois fixé à l'avance.

Tâche

Partant de l'état i , l'état suivant $i+1$ de l'automate est calculé d'après les règles suivantes en se basant uniquement sur l'état i .

- Si la cellule de l'état i contient un chiffre pair, alors dans l'état $i+1$ ce chiffre est incrémenté de 1 et se retrouve à droite et ceci d'autant d'espaces que sa valeur numérique. Dans l'exemple 2 ci-dessous, le premier chiffre 2 de l'état 0 se retrouve transformé en 3 à la deuxième cellule à droite du 2 ; le chiffre 6 de l'état 1 se transforme en 7 dans l'état 2 et est placé à la sixième position à droite de la position du 6 dans l'état précédent.
- Si la cellule de l'état i contient un chiffre impair, alors dans l'état $i+1$ ce chiffre est incrémenté de 1 et se retrouve à gauche et ceci d'autant d'espaces que sa valeur numérique. Dans l'exemple 2 ci-dessous, le chiffre 5 de l'état 0 se retrouve transformé en 6 à la cinquième cellule à gauche du 5; le chiffre 3 de l'état 0 se transforme en 4 dans l'état 1 et est placé à la quatrième position à gauche de la position du 3 dans l'état précédent.
- Si la cellule de l'état i contient 0, alors dans l'état $i+1$ ce chiffre devient 1 et sa position ne change pas.
- Si plusieurs chiffres arrivent sur la même cellule, ces chiffres s'additionnent. Dans l'exemple 2 ci-dessous, dans la suite 2311 de l'état 0, le 2 et le dernier 1 se retrouvent à l'état 1 sur la même cellule, le 2 devenant un 3 et le 1 un 2, la somme faisant bien 5 dans l'état 1.
- Lors du cas précédent, il se peut que cette addition produise un résultat supérieur à 9. Il faut alors décomposer ce résultat en ses chiffres et faire glisser ces chiffres dans les cellules adjacentes à droite en les additionnant avec les contenus existants.

Exemple:

```
66..220..33   État  $i$ 
.....164..   État  $i+1$ 
```

Voici en détail comment on obtient le résultat à l'état $i+1$:

Situation initiale:

1	2	3	4	5	6	7	8	9	10	11
6	6	.	.	2	2	0	.	.	3	3

Le 6 de la cellule 1 devient un 7 dans la cellule 7:

1	2	3	4	5	6	7	8	9	10	11
.	7

Le 6 de la cellule 2 devient aussi un 7 dans la cellule 8:

1	2	3	4	5	6	7	8	9	10	11
.	7	7	.	.	.

Le 2 de la cellule 5 devient un 3 qui s'ajoute au 7 de la cellule 7 pour donner 10:

1	2	3	4	5	6	7	8	9	10	11
.	10	7	.	.	.

Le 2 de la cellule 6 devient un 3 qui s'ajoute au 7 de la cellule 8 pour donner 10:

1	2	3	4	5	6	7	8	9	10	11
.	10	10	.	.	.

Le 0 de la cellule 7 devient un 1 qui s'ajoute au 10 de la cellule 7 pour donner 11:

1	2	3	4	5	6	7	8	9	10	11
.	11	10	.	.	.

Le 3 de la cellule 10 devient un 4 qui s'ajoute au 11 de la cellule 7 pour donner 15:

1	2	3	4	5	6	7	8	9	10	11
.	15	10	.	.	.

Le 3 de la cellule 11 devient un 4 qui s'ajoute au 10 de la cellule 8 pour donner 14:

1	2	3	4	5	6	7	8	9	10	11
.	15	14	.	.	.

Sous cette forme, les nombres résultants supérieurs à 9 doivent être décomposés en leurs chiffres et additionnés de gauche à droite. Ainsi le 15 de la cellule 7 est décomposé en ses chiffres 1 et 5 et le 14 de la cellule 8 en ses chiffres 1 et 4. Le 1 de 15 reste dans la cellule 7 alors que le 5 glisse dans la cellule 8 ; le 1 du 14 de la cellule 8 reste dans la cellule 8 et le 4 glisse dans la cellule 9. Une addition est alors effectuée de la gauche vers la droite. Dans cet exemple, on obtient ainsi:

```
.....15...
.....14..
```

En additionnant de la gauche vers la droite, on obtient finalement:

```
.....164..
```

¹ <http://www.cetteadressecomportecinquantesignes.com/AutomateNBR01.htm>

Restrictions

- $E = \{0; 1; 2; 3; 4; 5; 6; 7; 8; 9; .\}$ Symboles autorisés.
 $1 \leq N \leq 500$ Le nombre maximal d'itérations.
 $MAX = 80$ Longueur maximale de la ligne entrée.

Entrée et sortie du programme

- Le programme lit à partir de la console un entier N représentant le nombre d'itérations à effectuer. Il lit ensuite une ligne composée d'au moins un et au maximum MAX caractères de l'ensemble E . Cette ligne est la configuration de départ de l'automate.
- Le programme affiche à la console l'état de l'automate pour chaque itération. L'itération 0 est la configuration de départ. L'état de l'automate est décrit par une ligne débutant avec un entier i cadré à gauche sur 3 positions suivi d'un caractère '!', d'un espace et des MAX caractères décrivant l'état de l'automate à l'itération i . Si lors du calcul de l'état de l'automate le programme doit accéder à une cellule hors des limites des MAX cellules prévues, le programme affiche le message «Hors limites» et s'arrête.

Exemples d'exécution

Exemple 1

```
10
....6
0: ...6.....
1: .....7.....
2: ...8.....
3: .....9.....
4: ..10.....
5: .2.1.....
6: ..23.....
7: 4...3.....
8: .4..5.....
Hors limites
```

Exemple 2

```
50
.....25.....2311.....23
0: .....25.....2311.....23.....
1: .....6...3...4..25...4..3.....
2: .....4..7...6...53...4..5.....
3: .....8.....5..6..4..7..6...5.....
4: .....6...9..8.....75..6...7.....
5: .....10.....78..6...98.....7.....
6: ....2.1.8.....10.....978.....9.....
7: .....23.....21108.....10.....9.....
8: ...4...3.....2231.....2.110.....
9: ....4..5....4..53.....25.1.....
10: ..6....5..6..4..5..6....5.....
11: ...6...7...6...75..6...7.....
12: ..8.....78..6...78.....7.....
13: ...8.....9..8.....978.....9.....
14: .10.....19..8.....19.....9.....
15: 2.11....2.10....2.19.....
16: .25.....2311.....23.....
Hors limites
```

Remettez le programme sous le nom SOUPAUTOMAT{.PAS; .C; .CPP; .JAVA}.