

<div><b>be-OI 2015</b></div> <div>Finale</div> <div>2 mai 2015</div>		
--	--	--

<b>Olympiade belge d'Informatique</b> (durée : 2 x 3h maximum)
--

**Notes générales (à lire attentivement avant de répondre à la question)**

1. Installez-vous à la **place** qui vous a été **attribuée** par les organisateurs.
2. Vous ne pouvez avoir que de quoi écrire avec vous, les calculatrices, GSM,... sont **interdits**. Laissez toutes vos affaires à l'endroit indiqué par les surveillants.
3. Vous **ne pouvez** à aucun moment **communiquer** avec qui que ce soit, excepté avec les surveillants ou les organisateurs. Toute question portant sur la compréhension de la question ou liée à des problèmes techniques ne peut être posée qu'aux organisateurs ou via la plateforme de soumission (voir page suivante). Toute question logistique peut être posée aux surveillants.
4. Vous **n'avez pas** accès à Internet durant l'épreuve. Toute tentative de communication avec d'autres participants ou toute autre personne extérieure sera sanctionnée.
5. Vous **pouvez** utiliser toutes les fonctionnalités de la librairie standard du langage que vous aurez choisi (parmi Java ou C++) excepté au multi-threading et tout ce qui implique une communication avec le monde extérieur hors entrée et sortie standards. En pratique, vous ne pouvez donc pas accéder au réseau, ni lire ou écrire des fichiers sur le disque.
6. Vous pouvez demander des **feuilles de brouillon** aux surveillants.
7. Vous **ne pouvez pas quitter votre place** pendant l'épreuve. Si vous devez vous rendre aux toilettes, faites-en la demande à un surveillant. Ce dernier pourra décider d'accepter ou de refuser votre requête selon qu'il soit possible, ou pas, de vous accompagner tout en assurant la surveillance de l'épreuve.
8. Si vous finissez l'épreuve avant la fin, attendez l'autorisation d'un surveillant pour vous lever et quitter la salle.

**Spécifications techniques du concours**

1. Système d'exploitation : Ubuntu Linux 14.10 avec LXDE
2. Compilateur C++ : G++ 4.9.1, avec support C++-11
3. Compilateur Java : Oracle Java version 8u40
4. Editeur de texte par défaut : Gedit
5. Serveur de soumission : <http://final15.be-oi.be> (via un navigateur Internet)
6. Documentation C et C++ : via l'application DevHelp
7. Documentation Java : sur le bureau, dans le répertoire "docs". Décompressez l'archive et ouvrez `index.html` dans un navigateur.



Cette oeuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution 2.0 Belgique.

## Délivrables et calcul des points

Vous devez rendre votre programme via la système de soumission en ligne. Seul le code soumis sera utilisé pour calculer votre score final. Le code en tant que tel (sa qualité) ne sera pas utilisé par les évaluateurs.

Chaque tâche est séparée en sous-tâches chacune composée d'un ensemble de tests. Des algorithmes corrects mais non-optimaux peuvent ainsi marquer des scores partiels. Cependant vous ne marquez les points indiqués pour une sous-tâche que si tous les tests de celle-ci passent. Dans le cas contraire, vous ne marquez aucun point pour cette sous-tâche. Si une sous-tâche échoue, les sous-tâches suivantes sont tout de même évaluées. Les ensembles de tests sont construits de telle façon à ce qu'un même algorithme marque toujours le même nombre de points, quelque soit le langage utilisé.

Tout candidat finissant le concours (obtenant le score maximal) avant la fin de la session remportera **1 point supplémentaire par tranche de 6 minutes** restante avant la fin de la session. En cas d'égalité sur le score total de la finale, le score sans ces bonus sera pris en compte pour déterminer les positions finales. En cas de nouvelle égalité, la somme des heures de soumission du code vous ayant permis d'obtenir le score final pour chaque session sera utilisée.

## Instructions pratiques

Ces instructions détaillent comment travailler sur votre programme et ensuite soumettre ce que vous avez écrit sur le serveur officiel. Nous vous conseillons de lire ceci tout en l'appliquant à la *Tâche 0*.

### Charger et modifier les squelettes de programme

- La page *énoncé* de chaque tâche contient un fichier zip qui contient le squelette des programmes pour chacun des langages de l'épreuve. Téléchargez ce fichier dans le répertoire dans lequel vous souhaitez travailler.
- Faites un clic droit sur fichier zip puis *Extraire ici*. Dans le répertoire créé (par exemple Q0), vous trouverez un squelette de programme pour chaque langage de programmation (par exemple `task.cpp`), un exemple de données d'entrée (`input.txt`) et la sortie attendue pour cette entrée (`output-expected.txt`).
- Choisissez le langage dans lequel vous voulez travailler et double cliquez sur le fichier correspondant. Par défaut, il s'ouvre dans l'éditeur de code *gedit*. D'autres éditeurs sont installés si vous préférez. Nous vous recommandons de travailler sous *gedit* qui a été spécialement configuré pour l'épreuve.
- Le squelette fourni contient le code permettant de lire les données d'entrée du problème et d'écrire le résultat en sortie. Vous devez compléter ce squelette avec l'algorithme qui résout le problème.

### Compiler, exécuter et tester votre programme (seulement pour les tâches non-interactives)

- Lorsque vous êtes dans *gedit*, pressez **Ctrl+R** pour tester le programme que vous éditez. Cette commande sauve sur disque, compile, exécute et teste votre programme avec le fichier `input.txt` comme données. Les résultats s'affichent dans une console dans la partie inférieure de *gedit*.
- Si votre programme s'est compilé avec succès, votre programme s'exécute avec comme entrée le fichier `input.txt` présent dans le dossier. Si l'exécution s'est déroulée sans erreur, la sortie de votre programme se trouve dans le fichier `output.txt` et est comparée avec le résultat attendu se situant dans le fichier `output-expected.txt`. Dans tous les cas, si une erreur survient (compilation ou exécution), des informations seront affichées dans la console de *gedit*.
- Vous pouvez modifier `input.txt` pour effectuer d'autres tests, mais dans ce cas vous devez bien entendu adapter le fichier de sortie attendu `output-expected.txt`.

**Pour les tâches interactives** une compilation manuelle sera nécessaire. Cela sera expliqué dans la tâche elle-même.

### Soumettre votre programme

- Par sécurité et pour nous aider à plus facilement identifier l’auteur d’un code, indiquez votre nom en commentaire dans votre programme.
- La soumission se fait en envoyant votre fichier sur le serveur. Sur le serveur, rendez-vous sur la page *Soumissions* de la tâche correspondante. Cliquez sur *Parcourir* et sélectionnez le fichier dans lequel vous avez travaillé et ensuite *Soumettre*.
- Les résultats de votre soumission apparaissent sur cette même page dans la colonne *Score public*. Votre score final pour chacune des tâches sera le score maximal que vous avez obtenu pour cette question.
- Vous pouvez obtenir un détail de votre score par sous-tâche en cliquant sur *Détails* ainsi que la durée d’exécution pour chacun des tests.
- Seul le résultat de votre programme aux tests automatiques déterminera votre résultat, la qualité de votre code n’entre donc absolument pas en ligne de compte.

### Fonctionnement du système de soumission

- Lorsque vous soumettez un nouveau code, celui-ci est mis dans une file d’attente et sera exécuté plus ou moins vite selon le nombre d’autres participants ayant soumis récemment. Il est donc possible qu’en fin de séance, le temps à attendre pour que votre programme soit exécuté atteigne plusieurs minutes.
- Vous ne pouvez pas soumettre un code si vous en avez déjà soumis un il y a moins de **20 secondes**. Attention aux dernières minutes de l’épreuve !
- Si à la fin du temps de l’épreuve vous avez encore une soumission en attente, celle-ci sera bien entendu prise en compte. Par contre, vous ne saurez pas combien de points elle vous a rapportés.
- **Seul le meilleur score de chaque tâche sur toutes vos soumissions sera pris en compte pour calculer votre score final à la partie machine.**
- Avant de soumettre, faites bien attention à retirer tout message imprimé à la console que vous auriez ajouté pour *debugger* votre programme. Ceux-ci étant imprimés à la sortie standard, ils vont rendre le résultat de votre programme incorrect.

### Remarques

- Vous pouvez poser une question aux organisateurs du concours via la page *Communication* du serveur de soumission. Vous serez automatiquement notifié lorsqu’une réponse vous parviendra.
- Pour tout code non encore soumis sur la plate-forme de soumission, il peut être utile d’en faire une copie dans un second fichier pour prévenir les erreurs de manipulation. Ces « *backups* » sont de votre ressort.
- Le code dans *gedit* ne sera jamais évalué, seul le code soumis sur le serveur a de la valeur pour votre score final.
- **Problème connu :** Après avoir lancé le test de votre programme avec **Ctrl+R**, il peut arriver que le curseur de la souris se comporte comme si le test était toujours en cours. Il s’agit d’un bug de *gedit*. Le test est néanmoins bien terminé et vous pouvez continuer à travailler sereinement. Le curseur de *gedit* revient dans son état normal après un certain délai.
- **Problème connu :** Il est possible que certaines applications ou parties d’entre elles ne soient pas traduites en français et soit toujours en anglais. Si vous ne comprenez pas, demandez à un surveillant.
- Si un message concernant l’intégration de la souris est affiché en haut de votre écran, vous pouvez l’ignorer (le fermer), il s’agit d’un message lié à l’environnement virtuel dans lequel vous travaillez.
- Vous aurez peut-être remarqué une page *Tests* sur le serveur de soumission. Celle-ci vous permet de soumettre votre programme et un fichier d’entrée personnel pour qu’il soit évalué sur le même serveur que celui d’évaluation. Vous pouvez utiliser cette fonctionnalité si vous le souhaitez.

### Compilation manuelle

Si vous préférez utiliser un autre éditeur, vous pouvez utiliser les commandes suivantes pour compiler et tester vos programmes manuellement. Elle doivent être exécutée dans un terminal, dans votre répertoire de travail.

- Compilation C++ : `g++ -std=c++11 -Wall -Werror -o ./exec monprogramme.cpp`
- Compilation Java : `javac monprogramme.java`
- Exécution C/C++ : `./exec < input.txt > output.txt`
- Exécution Java : `java monprogramme < input.txt > output.txt`
- Test du résultat : `diff output.txt output-expected.txt` . Si cette commande ne retourne rien, cela signifie que les deux fichiers sont identiques.

### Première aide en cas de problème

- Si quelque chose n'est pas clair à propos de la description de tâche, utilisez le système de communication de la plate-forme pour poser la question aux organisateurs.
- Si votre ordinateur devient lent ou ne répond plus, appelez un surveillant avant qu'il puisse vous aider.
- En cas de handicap avéré suite à un problème de l'environnement de travail, les organisateurs pourront vous attribuer quelques minutes supplémentaires à la fin du concours.