

INFO0946 : Introduction à la Programmation

Interrogation Mi-Quadrimestre (durée : 1h00)

Benoit Donnet, Simon Liénardy

Consignes (à lire attentivement avant de répondre aux questions)

Suite à la crise sanitaire liée à la pandémie de Covid-19, l'interrogation de mi-quadrimestre est organisée à distance via une session **WebEx** (pour poser vos questions sur l'énoncé), via CAFÉ pour l'encodage de vos réponses (ainsi que la correction automatique) et la **Plateforme de Soumission** pour déposer vos réponses.

1. Le présent questionnaire et les différents fichiers annexes (i.e., le fichier servant de canevas pour la soumission de vos réponses et le squelette de code) sont disponibles sur **eCampus** (INFO0946, Sec. Interrogation Mi-Quadrimestre).
2. Contrairement aux Challenges, CAFÉ ne fera pas de correction automatique avec feedback sur vos réponses. La seule chose que vérifiera CAFÉ, c'est l'encodage de vos réponses dans le canevas. Un email vous sera envoyé automatiquement avec le résultat de ces tests. Si jamais il y a des erreurs d'encodage, vous avez la possibilité de charger de nouveau votre fichier (autant de fois que nécessaire, dans les limites du temps imparti par l'interrogation).
3. Il y a un fichier de canevas pour l'unique question de l'interrogation. Vous devez donc télécharger, depuis **eCampus** (Sec. Interrogation Mi-Quadrimestre), le fichier texte (`interro.txt`). Ce fichier doit être complété et placé dans une archive `.zip` pour la soumission (voir page suivante).
4. La correction de l'interrogation sera réalisée automatiquement par CAFÉ, après l'interrogation en elle-même. Il est, dès lors, important que votre code compile. Sinon, vous risquez une note nulle à la question!! Pour s'assurer de la compilation, nous vous fournissons aussi (voir **eCampus**, Sec. Interrogation Mi-Quadrimestre) un squelette de code que vous pouvez compléter, localement sur votre machine, afin de tester (au moins) la compilation de votre solution.
5. La **durée de l'interrogation est de 1h**. N'attendez pas la dernière seconde pour soumettre votre fichier.

Soumission

Un fichier servant de canevas pour la soumission de l'interro est disponible sur la page web du cours¹. Le nom du fichier est `interro.txt`. Le squelette pour cette interrogation est donc contenu dans le fichier `interro.txt`. Par la suite, libre à vous de modifier le nom du fichier que vous soumettez, cela n'a pas d'importance.²

Le fichier texte à soumettre doit être compressés en une archive « `.zip` ». Voici comment procéder sur les systèmes d'exploitation les plus courants.

Sous Windows Il suffit de cliquer sur le fichier à l'aide du bouton droit de la souris, sélectionner « Envoyer vers... » et sélectionner ensuite « Dossier compressé ».

Sous Linux (Ubuntu, Fedora, Linux Mint, ...) Il suffit de cliquer sur le fichier à l'aide du bouton droit de la souris, sélectionner « Compresser... ». Veillez bien à sélectionner « `.zip` » dans la liste des extensions possibles pour le fichier.

Sous OS X Cliquez sur le fichier en maintenant la touche Contrôle enfoncée (ou cliquez avec 2 doigts), sélectionnez « Compresser ».

Dans tous les cas Ne soumettez pas de fichier `.tar.gz`, `.7z`, `.rar` ou autre ! C'est bien un fichier `.zip` qui est attendu. Le nom de l'archive importe peu, tant que c'est une archive zip valide, dont le nom se termine bien par « `.zip` » **et ne comporte pas de caractères spéciaux comme des espaces, des parenthèses, etc.**

Si vous commettez un erreur dans la soumission, comme par exemple :

- Donner un mauvais nom à l'archive ou utiliser des caractères inhabituels (e.g., des parenthèses, accent, espace, ...);
- Soumettre deux fois d'affilée en cliquant trop rapidement;
- Mal placer les réponses dans le fichier `interro.txt`
- ...

C'est dommage pour vous³. Redoublez d'attention la prochaine fois ! Pour autant, la plateforme de soumission et le soucis d'équité entre tous les étudiants ne permettent pas de vous octroyer une nouvelle soumission.

1. [eCampus](#), Sec. Interrogation Mi-Quadrimestre.

2. Prenez soin, par contre, d'éviter les caractères particuliers (espace, accent, ...).

3. Nous partageons votre peine.

Agenda

Votre interrogation doit être soumise pour le **04/11, 09h30**, au plus tard. Dans le cadre de l'interrogation, vous disposez d'autant de soumissions que vous le souhaitez. Cependant, attention : CAFÉ ne vérifiera que le format de votre soumission (par conséquent, le feedback envoyé par email ne portera que sur le format). Votre production sera évaluée, par CAFÉ, après le 04/11, 09h30.

Une session **WebEx**⁴ a été mise en place afin de répondre à vos questions en cours d'interrogation. Il suffit de cliquer sur le lien **WebEx** et d'entrer vos prénoms/noms (pas la peine d'installer le client desktop sur votre machine). Si nécessaire, vous pourrez partager votre écran avec nous. Une fois la question résolue, vous pouvez quitter la session **WebEx**.

4. C'est un logiciel de vidéo-conférence

1 Construction par Invariant de Boucle – Énoncé du Problème

Dans ce problème, nous considérons des nombres entiers positifs. En particulier, les nombres a et b vous sont déjà fournis. On demande d’afficher, sur la sortie standard, **le nombre** de chiffres impairs composant chaque nombre de l’intervalle $[a, b]$.

Voici un exemple de trace d’exécution de votre programme, pour $a = 567$ et $b = 573$:

```
567 : 2
568 : 1
569 : 2
570 : 2
571 : 3
572 : 2
573 : 3
```

Votre code, doit afficher sur la sortie standard, une ligne par nombre dans $[a, b]$. Chaque ligne est composée du nombre considéré, suivi d’un espace, suivi de deux points, suivis d’un espace suivi du nombre de chiffres impairs dans la représentation décimale du nombre considéré (c.-à-d., faire exactement comme dans l’exemple). Un non-respect de cette contrainte d’affichage sur la sortie standard empêcherait CAFÉ de corriger votre code, menant de facto à une note nulle. Soyez donc très rigoureux!!

1.1 Squelette du Code

Le squelette de votre code est le suivant :

```
1 #include <stdio.h>
2
3 int main() {
4     unsigned int a = ...;
5     unsigned int b = ...;
6     const unsigned int A_INIT = a;
7     const unsigned int B_INIT = b;
8
9     //votre code viendra ici (variables + instructions)
10
11 }
```

Vous devez considérer, lors de la soumission, que votre extrait de code est copié et collé⁵ à l’endroit indiqué par le commentaire « Votre code viendra ici (variables + instructions) ». Dès lors, redéclarer les constantes `A_INIT`, `B_INIT`, ou encore les variables `a` et `b` fera échouer la compilation.

Il est impératif que votre code compile lors de la soumission, sous peine de mener à une cote nulle pour l’entièreté de l’interrogation. À cette fin, le squelette de code ci-dessus vous est fourni (voir [eCampus](#), Sec. Interrogation Mi-Quadrimestre). Libre à vous de le compléter avec votre code pour vérifier la bonne exécution de votre code.

Lors de votre soumission, vous devrez fournir votre code ainsi que les Invariants Graphiques et leurs Fonctions de Terminaison. La façon de formuler vos Invariants Graphiques et vos Fonctions de Terminaison sont décrites dans la suite de de document. Le fichier servant de canevas à votre soumission est disponible sur [eCampus](#) (Sec. Interrogation Mi-Quadrimestre).

5. En fait, c’est à quelques détails près le cas.

1.2 Contraintes à Respecter

Attention, pour résoudre ce problème, vous devez respecter ces contraintes. Ne pas le faire peut entraîner jusqu'à une note nulle pour l'interrogation :

- Votre solution devra être construite uniquement autour de boucles **while** ;
- Votre solution doit respecter fidèlement le format d'affichage sur la sortie standard.

Vous ne devez **surtout pas** :

- Redéclarer les constantes `A_INIT`, `B_INIT` ainsi que les variables `a` et `b` ;
- Écrire `int main() { »` ni la `« }` qui correspond à celle ouverte après `main()` .
- Soumettre un code qui contient un appel à une fonction différente de `printf` ;
- Soumettre un code qui contient un opérateur du langage C non vu au cours ;
- Utiliser une instruction du langage C non vue au cours ou faisant preuve d'une mauvaise réflexion. Par exemple `break`, `goto`, ou encore `continue` ;
- Utiliser un tableau ou une structure.

2 Spécifier un Invariant Graphique

2.1 Introduction : Découpe en SP

Nous avons vu au cours une approche méthodologique que nous vous encourageons à mettre en pratique dans tous vos cours. La résolution du problème de l'interrogation ne fait donc pas exception.

Puisqu'il est très difficile d'évaluer et de coter automatiquement une découpe en SP, nous vous proposons notre découpe, qui a l'avantage d'être simple et directe :

SP₁ : énumération dans $[a, b]$ et affichage de la quantité de chiffres impairs pour chaque valeur de l'intervalle ;

SP₂ : calcul du nombre de chiffres impairs pour un nombre donné.

L'enchaînement est le suivant :

$$SP_2 \subset SP_1$$

Dans la suite, nous vous proposons de compléter les Invariants Graphiques qui se rapportent à ces deux SP. La manière d'encoder votre solution dans le fichier de soumission sera également précisée.

2.1.1 Conseils

Il est recommandé de d'abord rechercher les Invariants Graphiques **sans** s'inspirer des canevas imposés. Par exemple, vous pouvez faire votre propre dessin sur une feuille de papier ou en utilisant le **GLI** et par la suite, comparez ce dessin aux Invariants Graphiques à compléter, détaillés dans les sections suivantes. De toute manière, si votre Invariant Graphique est correct, il devrait contenir la *même information* que celui que vous soumettrez selon notre modèle et vous n'aurez qu'à adapter légèrement votre solution pour la rendre compréhensible par la machine.

Si vous êtes bloqué·e, un coup d'œil au modèle devrait vous mettre sur la piste.

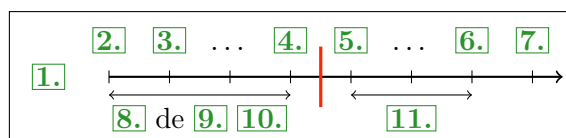
2.2 Invariant Graphique du **SP₁**

2.2.1 Schéma Muet à Compléter

Soit le schéma muet ⁶, suivant ⁷ :

6. c'est-à-dire expurgé de toute notation

7. Tout ce qui est dans le cadre fait partie de l'Invariant Graphique.



Les boîtes **2.** à **7.** servent à mentionner des indices. Ces boîtes doivent donc être remplacées par des constantes ou des noms de variables.

Toutes les boîtes ne doivent pas forcément être précisées : si vous indiquez que la valeur de la boîte **2.** est « k », il est implicite que la valeur de la boîte **3.** est « $k + 1$ » : inutile alors de le préciser.

Les boîtes **1.**, **8.** à **11.** doivent être remplacées par des mots. Les différents choix possibles sont repris dans la table 1 :

Pour la boîte 1.	Pour la boîte 8.	Pour la boîte 9.	Pour la boîte 10.	Pour la boîte 11.
1. \mathbb{R}	1. mots	chiffres négatifs	calculés	à afficher
2. Maître Gims	2. guitares	chiffres pairs	affichés	à parcourir
3. \mathbb{Z}	3. nombres	chiffres impairs	compressés	à compresser
4. ligne	4. espaces	chiffres positifs	décidés	à effectuer
5. \mathbb{N}	5. assistants	chiffres premiers	effectués	à décider
6. \mathbb{C}	6. phrases	chiffres derniers	parcourus	à trouver

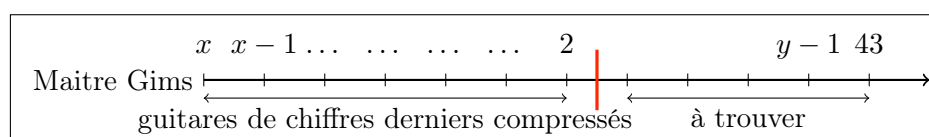
TABLE 1 – Invariant de Boucle : Possibilités de choix pour les boîtes **1.**, **8.** à **11.**.

Attention, nous inspecterons votre code pour détecter s’il respecte l’Invariant Graphique ainsi formé.

2.2.2 Encodage dans le Fichier de Soumission

Veuillez indiquer, dans le fichier de soumission, à la réponse sur l’Invariant Graphique, le numéro de la boîte, suivi d’un point, suivi de la valeur numérique ou du nom de variable ou constante de votre choix (pour les boîtes de **2.** à **7.**) ou d’un nombre entre 1 et 6 correspondant à votre choix (pour les boîtes **1.**, **8.** et **11.**).

Si vous pensez qu’un bon Invariant Graphique serait ⁸ :



Alors, complétez le fichier de réponse comme suit :

1. 2
2. x
3. x-1
4. 2
5. _
6. y-1
7. 43
8. 2

⁸. Ce n’est toujours malheureusement pas le cas...

9. 6
10. 3
11. 6

Vous voyez qu'il est possible d'affecter une variable ou une constante d'un modificateur +1 ou -1. Il suffit dans ce cas d'écrire ce +1 ou -1 comme montré ci-dessus. N'introduisez pas de parenthèses.

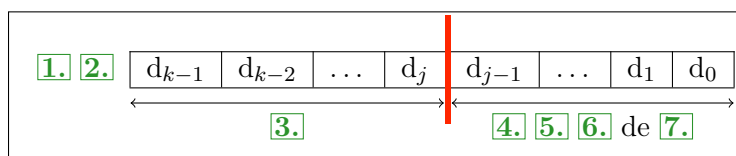
Dans tous les cas, si vous pensez que la bonne réponse consiste à ne rien écrire à l'emplacement d'une boîte, n'inscrivez rien comme réponse ou un « _ ».

Pour votre facilité, (et pour les distraits qui en oublieraient un), les numéros des 11 boîtes sont déjà inscrits dans le fichier de réponse.

2.3 Invariant Graphique du \mathbf{SP}_2

2.3.1 Schéma Muet à Compléter

Soit le schéma muet⁹, suivant¹⁰ :



Dans ce schéma, chaque valeur d_i (avec $0 \leq i \leq k-1$) représente un symbole dans $\{0, \dots, 9\}$.

Les boîtes **2.** et **4.** servent à mentionner des indices. Ces boîtes doivent donc être remplacées par des constantes ou des noms de variables.

Les boîtes **1.**, **3.** et **5.** à **7.** doivent être remplacées par des mots. Les différents choix possibles sont repris dans la table 2 :

Pour la boîte 1.	Pour la boîte 3.	Pour la boîte 5.	Pour la boîte 6.	Pour la boîte 7.
1. mot	1. à afficher	1. contient	1. le mot	1. chiffres négatifs
2. guitare	2. à parcourir	2. emballe	2. la guitare	2. chiffres pairs
3. nombre	3. à compresser	3. emmène	3. le nombre	3. chiffres impairs
4. espace	4. à calculer	4. attend	4. l'espace	4. chiffres positifs
5. assistant	5. à décider	5. calcule	5. l'assistant	5. chiffres premiers
6. phrase	6. à trouver	6. énumère	6. la phrase	6. chiffres derniers

TABLE 2 – Invariant de Boucle : Possibilités de choix pour les boîtes **1.**, **3.** et **5.** à **7.**.

Attention, nous inspecterons votre code pour détecter s'il respecte l'Invariant Graphique ainsi formé.

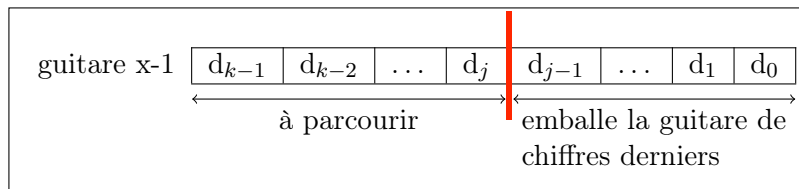
9. c'est-à-dire expurgé de toute notation

10. Tout ce qui est dans le cadre fait partie de l'Invariant Graphique.

2.3.2 Encodage dans le Fichier de Soumission

Veuillez indiquer, dans le fichier de soumission, à la réponse sur l'Invariant Graphique, le numéro de la boîte, suivi d'un point, suivi de la valeur numérique ou du nom de variable ou constante de votre choix (pour les boîtes de **2.** et **4.** ou d'un nombre entre 1 et 6 correspondant à votre choix (pour les boîtes **1.**, **3.** et **5.** et **7.**).

Si vous pensez qu'un bon Invariant Graphique serait ¹¹ :



Alors, complétez le fichier de réponse comme suit :

1. 2
2. x-1
3. 3
4. _
5. 2
6. 2
7. 6

Vous voyez qu'il est possible d'affecter une variable ou une constante d'un modificateur +1 ou -1. Il suffit dans ce cas d'écrire ce +1 ou -1 comme montré ci-dessus. N'introduisez pas de parenthèses.

Dans tous les cas, si vous pensez que la bonne réponse consiste à ne rien écrire à l'emplacement d'une boîte, n'inscrivez rien comme réponse ou un « _ ».

Pour votre facilité, (et pour les distraits qui en oublieraient un), les numéros des 7 boîtes sont déjà inscrits dans le fichier de réponse.

3 Fonction de Terminaison

La Fonction de Terminaison ¹² permet de fournir la preuve que la boucle se termine. Dans cette interrogation, nous vous demandons de fournir la Fonction de Terminaison de vos deux boucles (puisque 1 Invariant de Boucle == 1 boucle).

Dans le fichier de réponses, aux questions sur la Fonction de Terminaison, nous vous demandons de nous la fournir sous la forme d'une **expression C** valide.

La correction s'assurera entre autres :

- Que cette expression utilise des variables de votre code ;
- Que son domaine est bien l'ensemble des Entiers (positifs si le gardien est vrai) ;
- Que sa valeur décroît strictement entre deux itérations.

11. Ce n'est toujours malheureusement pas le cas...

12. À ne confondre ni avec le Critère d'Arrêt, ni avec le Gardien de Boucle

3.1 Exemple

Si vous pensez que la Fonction de Terminaison de votre code, qui manipule la variable `toto` et la constante `G` devrait être :

$$t = G + 17 - \text{toto}$$

Encodez :

`G + 17 - toto`

N'indiquez pas « `F =` » ou « `t =` » mais seulement l'expression qui sert à calculer la valeur de votre Fonction de Terminaison. De plus, n'ajouter pas « `> 0` » à votre soumission. en effet, $G + 17 - \text{toto} > 0$ n'est pas une fonction à valeur entière mais Booléenne¹³.

13. Et donc c'est incorrect...