

# INFO0946 : Introduction à la Programmation

## Challenge 1 (Instructions Itératives)

Benoit Donnet, Géraldine Brieven

### 1 Énoncé du Problème

Un *data center* (DCN) est “un lieu (et un service) regroupant des équipements constituant le système d’information d’une ou plusieurs entreprises (ordinateurs centraux, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.). Il peut être interne et/ou externe à l’entreprise, exploité ou non avec le soutien de prestataires. Il fournit des services informatiques en environnement contrôlé (climatisation) et sécurité (système anti-incendie, contre le vol et l’intrusion, etc.), avec une alimentation d’urgence et redondante.”<sup>1</sup>

Dans cet exercice, on s’intéresse à l’évolution de la consommation électrique d’un DCN. Sachant que cette consommation augmente de 8% par an, après combien d’années la consommation aura doublé étant donné une certaine consommation initiale ?

Le squelette de votre code est le suivant :

```
1 #include <stdio.h>
2
3 int main() {
4     float consommation = ...;
5     const float CONSOMMATION_INIT = consommation;
6     int annees;
7
8     //votre code viendra ici (variables + instructions)
9
10    printf("%d_--_d_--_d_annees\n", (int)CONSOMMATION_INIT, (int)consommation, annees);
11 } //fin programme
```

La variable `consommation` correspond à une variable contenant la consommation électrique du DCN. Vous devez considérer, lors de la soumission, que cette variable est déjà initialisée à une certaine valeur positive correspondant à la consommation initiale du DCN. La variable `annees` (non initialisée – c’est à vous de le faire dans votre solution) sera celle qui contiendra le résultat de votre calcul. La constante `CONSOMMATION_INIT` n’a que pour seul objectif de conserver la valeur d’origine de `consommation` afin de faciliter l’affichage final. Attention, la ligne 10 du squelette contient du transtypage afin de faciliter la correction automatique.

Lors de votre soumission, vous devrez fournir trois versions de votre code. La première version utilise une boucle de type `while`, la seconde une boucle de type `for`. Enfin, la dernière version se base sur une boucle de type `do ... while`. Chacune des versions ne doit contenir qu’une et une seule boucle.

Pour chaque version, vous ne devez fournir que l’**extrait** de code qui remplace la ligne 8 du squelette ci-dessus : « //votre code viendra ici ». En particulier **ne** recopiez **pas** la commande `printf` de la ligne 10 du squelette.

#### 1.1 Critères éliminatoires

Le non respect d’une de ces consignes<sup>2</sup> entraînera une cote nulle pour la boucle qui l’enfreint :

- Redéclarer une des variables (`consommation`, `CONSOMMATION_INIT`, `annees`);
- Proposer autre chose qu’une seule boucle de celle du type attendu;
- Soumettre un code qui contient un appel à une fonction;

1. cfr. [https://fr.wikipedia.org/wiki/Centre\\_de\\_données](https://fr.wikipedia.org/wiki/Centre_de_données)

2. Certaines semblent évidentes, d’autres mentionnent des éléments du langage C non (encore) vus au cours.

- 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 Agenda

Votre challenge doit être soumis pour le **vendredi 01/10, 20h00**, au plus tard. Pour rappel, vous disposez de maximum trois essais.

## 3 Soumettre une Archive **.zip**

Pour tous les challenges, un fichier servant de canevas pour la soumission du challenge est disponible sur la page web du cours<sup>3</sup>. Le nom du fichier est `challengeX.txt` où X est remplacé par le numéro du challenge. Le squelette pour ce challenge 1 est donc contenu dans le fichier `challenge1.txt`. Par la suite, libre à vous de modifier le nom du fichier que vous soumettez, cela n'a pas d'importance.

Tous les challenges doivent 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) ;
- Soumettre deux fois d'affilée en cliquant trop rapidement ;
- Mal placer les réponses dans le fichier `challenge1.txt`
- ...

C'est dommage pour vous<sup>4</sup>. 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.

---

3. **eCampus**, Sec. Challenges.

4. Nous partageons votre peine.