

INFO0946 : Introduction à la Programmation

Challenge 1 (Instructions Itératives)

Benoit Donnet, Simon Liénardy

1 Énoncé du Problème

Pour multiplier entre eux deux entiers non négatifs, x et y , il est possible de procéder de la façon suivante :

- à chaque étape, on divise x par 2 (division entière, sans tenir compte d'un éventuel reste) et on multiplie y par 2 et ceci jusqu'à ce que la valeur de x soit égale à 1 ;
- la valeur de $x \times y$ est alors la somme des valeurs de y correspondant à des x impairs.

Par exemple, $17 \times 13 = 221$. Soit :

17	13	←	
8	26		
4	52		
2	104		
1	208	←	on a bien $221 = 13 + 208$

Le squelette de votre code est le suivant :

```
1 #include <stdio.h>
2
3 int main() {
4     unsigned int x = ...;
5     unsigned int y = ...;
6     const unsigned int X_INIT = x;
7     const unsigned int Y_INIT = y;
8     unsigned int multiplication;
9
10    //votre code viendra ici (variables + instructions)
11
12    printf("%u*%u=%u--%u\n", X_INIT, Y_INIT, multiplication, X_INIT * Y_INIT);
13 } //fin programme
```

Les variables x et y correspondent aux variables pour lesquelles vous voulez effectuer la multiplication. Vous devez considérer, lors de la soumission, que chacune de ces deux variables est déjà initialisée à une certaine valeur positive. La variable `multiplication` (non initialisée – c'est à vous de le faire dans votre solution) sera celle qui contiendra le résultat de votre calcul, suivant l'algorithme décrit ci-dessus. Les constantes `X_INIT` et `Y_INIT` n'ont que pour seul objectif de conserver les valeurs d'origine de x et y afin de faciliter l'affichage final.

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 version 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 consignes¹ entrainera une code nulle pour la boucle qui l'enfreint :

- Redéclarer une des variables (`x`, `y`, `X_INIT`, `Y_INIT`, `multiplication`);
- Proposer autre chose qu'une seule boucle de celle du type attendu;
- Soumettre un code qui contient un appel à une fonction;
- 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 09/10, 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 cours². 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 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. Certaines semblent évidentes, d'autres mentionnent des éléments du langage C non (encore) vus au cours.

2. <http://www.ecampus.ulg.ac.be>, Sec. Challenges.

3. Nous partageons votre peine.