

EXERCICE 07 – ITÉRATIVES 2

Préparé par Benjamin Lemelin et Pierre Poulin le 18 janvier 2023

1 Travail à effectuer

Ouvrez le projet fourni avec cet énoncé. Il devrait déjà contenir un fichier **Program.cs**.

1.1 Lire un nombre

Localisez la fonction **ReadNumber**. Dans cette fonction, lisez un nombre entre 0 et 10 (inclusivement) à partir de la console. Pour ce faire, utilisez l'extrait de code déjà fourni dans la fonction. Assurez-vous de bien le comprendre en lisant les commentaires.

La fonction doit demander un nombre à l'utilisateur jusqu'à ce qu'il entre un nombre valide et entre 0 et 10 inclusivement. Vous aurez donc compris qu'il faudra écrire une boucle.

1.2 Lire un mot de passe

Inspirez-vous du code que vous avez fait à la section précédente pour lire un nom d'utilisateur et un mot de passe dans la fonction **ReadUsernamePassword**. La seule combinaison de nom d'utilisateur et de mot de passe acceptée est **intro** et **w10**.

Si l'authentification réussit, le programme doit afficher **Authentification réussie**. Sinon, il doit afficher **Authentification échouée**.

Information

N'oubliez pas de faire usage de constantes lorsque pertinent.

1.3 Limiter le nombre de tentatives

Localisez la fonction **ReadUsernamePasswordLimitedAttempts**. Cette fonction fait la même chose que la précédente, à la différence qu'elle donne 3 chances à l'utilisateur d'entrer la bonne combinaison.

Prudence

Une sorte particulière de boucle est beaucoup mieux indiquée que les autres pour faire ce numéro.

Astuce

Personne n'aime les constantes déclarées en double. Déplacez les constantes dans la portée supérieure (soit la portée globale).

1.4 Nombres aléatoires

Il arrive fréquemment en programmation que nous ayons à générer des nombres aléatoires. C'est tout particulièrement important dans les jeux vidéo afin de créer de la variété. En C#, cela se fait ainsi :

```
Random random = new Random();  
int value = random.Next(1, 100); // Génère un nombre aléatoire entre 1 et 99.
```

Pour générer plusieurs nombres aléatoires, vous pouvez tout simplement utiliser une boucle :

```
Random random = new Random();  
while (true)  
{  
    int value = random.Next(1, 100);  
}
```

Localisez la fonction **GenerateRandomNumbers**. Cette dernière doit générer 3000 nombres aléatoires entre 1 et 99 et compter leur répartition.

Information

Normalement, un *bon* générateur de nombre aléatoire, sur une grande quantité de nombres, générer autant de nombres entre 1 et 33 que de nombres entre 34 et 66 ou qu'entre 67 et 99. Par exemple, si l'on génère 3000 nombres aléatoires entre 1 et 99, il devrait y en avoir environ 1000 entre 1 et 33, environ 1000 entre 34 et 66 et environ 1000 entre 67 et 99.

2 Modalités de remise

Remettez votre projet Visual Studio sur LÉA, dans la section travaux, à l'intérieur d'une archive *Zip*. Supprimez tous les dossiers temporaires, à savoir les dossiers **.vs**, **TestResults**, **bin** et **obj**.