

## InitDev1 - Contrôle commun TP C++

### Durée : 1h30 (2h pour les 1/3 temps)

- Vous devez déposer **deux** fichiers **main.cpp** sur Moodle, un par exercice, déposé dans l'item correspondant du test.
- La durée totale est paramétrée sur Moodle, aucun rendu ne sera possible au-delà de 1h30 (2h pour les tiers-temps), même si vous vous êtes trompés de fichier.
- Déposez une version préliminaire de vos fichiers au moins une demi-heure avant la fin.

### Exercice 1 : Boucles (environ 4 points, 20 minutes)

On souhaite réaliser une application permettant à un enfant de réviser ses tables de multiplications. Compléter le programme suivant pour demander un nombre positif  $n$  plus petit que 10 (on ne fera pas de vérification). Demander ensuite un nombre multiple de  $n$ . Le programme vérifie que l'enfant a bien donné un multiple<sup>1</sup> de  $n$ . Si ce n'est pas le cas, il lui redemande de faire une tentative. L'enfant a 4 essais au plus. À la fin, si l'enfant a trouvé un multiple, le programme le félicite en lui précisant le nombre de tentatives qu'il a faites, sinon il affiche un message d'encouragement lui suggérant de recommencer plus tard. (Voir les exemples d'exécution ci-dessous.)

Utiliser les constantes fournies afin de faciliter vos tests. Il ne vous est pas demandé de rendre votre jeu d'essais pour cet exercice.

```
#include <iostream>
using namespace std;

const int SUP = 10;
const int MAXESSAIS = 4;

int main()
{
```

#### Exemple 1 d'exécution du programme :

```
> Donne-moi un nombre positif plus petit que 10
> 7
> Bien. Maintenant donne-moi un multiple de 7 :
> 18
> 18 n'est pas un multiple de 7. Essaie encore :
> 42
> Bravo ! Tu as trouvé un multiple de 7 en 2 essais.
```

#### Exemple 2 d'exécution du programme :

```
> Donne-moi un nombre positif plus petit que 10
> 9
> Bien. Maintenant donne-moi un multiple de 7 :
> 19
> 19 n'est pas un multiple de 9. Essaie encore :
> 28
> 28 n'est pas un multiple de 9. Essaie encore :
> 34
> 34 n'est pas un multiple de 9. Essaie encore :
> 42
> Dommage, 42 n'est toujours pas un multiple de 9. Relance le programme quand tu
voudras rejouer !
```

---

<sup>1</sup> **Rappel** : un nombre  $i$  est multiple de  $n$  si  $i$  modulo  $n$  (noté  $i \% n$ ) est nul, c'est-à-dire si le reste de la division entière de  $i$  par  $n$  est nul.

**Exercice 2 – vecteurs (environ 16 points, 1 heure)**

- Séparer clairement par un commentaire les réponses aux différentes questions.
- Chaque question doit être traitée indépendamment
- Veiller à faire les affichages nécessaires au test du bon fonctionnement de votre programme.

Démarrer un programme en déclarant le vecteur suivant :

```
vector<int> v1 = {-2, 3, 7, 1, 2, 3, 7, 3, -2, 7, 0};
```

**Attention**, le contenu de `v1` donné ci-dessus n'est qu'un exemple. Votre programme doit fonctionner avec n'importe quel vecteur, même vide.

1. Afficher les valeurs du vecteur **v1** les unes à côté des autres, séparées par des virgules. Il ne doit pas y avoir de virgule en fin de ligne.
2. Déclarer un vecteur **v2** et le remplir avec les valeurs de **v1** dans l'ordre inverse.
3. Déterminer et afficher la plus grande valeur du vecteur **v1** ainsi que le nombre de fois où elle apparaît. Par exemple, pour le vecteur donné ci-dessus, on affichera :

```
La valeur la plus grande est 7 et elle apparait 3 fois.
```

Attention : effectuer le traitement avec un seul parcours du vecteur.

4. Demander à l'utilisateur de saisir une valeur **val** et un nombre positif **n** (on ne fera pas de vérification) et afficher si **val** est présente au moins **n** fois dans le vecteur **v1** ou pas. Par exemple, pour le vecteur donné ci-dessus avec la valeur **val** égale à 3 et le nombre **n** égal à 2, on affichera :

```
La valeur 3 apparait au moins 2 fois
```

Attention : le traitement doit s'arrêter dès qu'une réponse peut être donnée. Utiliser la boucle la plus appropriée pour ce cas de figure.

5. Constituer un jeu d'essais complet pour la question 4. Veiller à bien préciser, pour chaque cas : le test abstrait, le test concret et le résultat attendu. Pour le test concret, vous pouvez utiliser le nom de vecteur `v1` sans avoir à préciser son contenu. Mettre le jeu d'essais en commentaire à la fin du fichier.

***Vérifiez que vous avez bien déposé les deux fichiers `main.cpp` sur Moodle.***