Tableaux à une dimension TP 1

Initiation au C++
20 points

Partie 1 : Hello world !

1) Connecte toi à ton compte repl.it.
   Crée un nouveau repl en choisissant le langage C++. Nomme-le **tableaux TP 1**.
   Créer un nouveau repl ===> à 2'50 dans cette **vidéo**.

2) Renomme le fichier **main.cpp** en **hello.cpp**.
   Renommer un fichier ===> à 3'45 dans cette **vidéo**.

3) Modifie le code de ce fichier de façon à obtenir le programme ci-dessous :

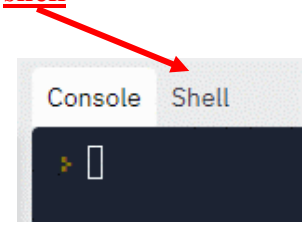
```cpp
//Ceci est un commentaire d'une seule ligne

/* Ceci est un commentaire
de plusieurs lignes */

#include <iostream>
using namespace std ;

int main() {
  cout << "Hello World!" << endl ;
  return 0 ;
}
```

4) Exécute ce programme :
- en cliquant sur l'onglet **shell**



- en tapant cette instruction qui **compile** ton programme et génère un fichier exécutable nommé **hello**

```
~/Tableaux-TP-1$ g++ hello.cpp -o hello
~/Tableaux-TP-1$
```

- en tapant enfin cette instruction qui **exécute** ton programme :

```
~/Tableaux-TP-1$ g++ hello.cpp -o hello
~/Tableaux-TP-1$ ./hello
```

5) Associe à chacune des instructions C++ l'instruction Python équivalente :

Instructions C++
#include
int main( )
cout << … << endl ;

Instructions Python
print
import
if __name__ == "__main__":

Clique ici si tu ne vois pas du tout comment faire **solution**

A retenir :

En Python, chaque bloc de code était défini par son indentation. **En C++, chaque bloc de code sera délimité par des accolades { }.** En C++, l'indentation n'est pas nécessaire à la bonne exécution du code, elle est juste là pour faciliter sa lecture par un être humain.
A l'intérieur de la partie principale du programme, chaque ligne de code devra se terminer par un point-virgule.
La structure de tes programmes sera donc la suivante :

```
1   #include <iostream>
2   using namespace std ;
3
4   int main() {
5
6     // Tu taperas ton code ici
7
8     return 0 ;
9   }
```

Je m'évalue

Exercice 1 (2 points)

1) Crée un nouveau fichier que tu nommeras **partie1_ex1.cpp**.
   Créer un nouveau fichier ===> à 4'10 dans cette **vidéo**.

2) Écris un programme qui affiche, sur la console de sortie :
   J'apprends le C++.
   Je vais faire des erreurs mais c'est en faisant des erreurs que j'apprendrai à programmer.

Exercice 2 (3 points)

1) Crée un nouveau fichier que tu nommeras **partie1_ex2.cpp**.

2) Tape le programme ci-dessous.

```
1   include <iostream>
2   using namespace std
3
4   int main {
5
6     cout >> "J'apprends le C++." >> endl
7
8     return 0 ;
9   }
```

3) Corrige le code pour qu'il compile correctement puis exécute-le.

Partie 2 : les variables

Nous avons vu qu'en Python, pour créer une nouvelle variable, il suffit d'écrire son nom. Le langage Python ne nécessite pas de déclarer le type de la variable (int, float, etc.).
En C++, en revanche, il est nécessaire de le faire.
On indique les choses exactement dans cet ordre : **Type Nom (Valeur)**
Correspondance des types entre le langage C++ et le langage Python :

Quelques types en C++	Types correspondant en Python
int	int
double	float
string	str

Crée un nouveau fichier et nomme-le **declaration.cpp**.
Tape le code ci-dessous dans ce fichier. Compile le puis exécute le. Regarde bien comment il fonctionne.

```
1   #include <iostream>
2   using namespace std ;
3
4   int main() {
5       int ageUtilisateur(16) ;
6       cout << "Vous avez aux environs de " << ageUtilisateur << " ans." << endl ;
7
8       double temperature(37.2) ;
9       cout << "Votre température corporelle est environ de " << temperature << " °C." <<
        endl ;
10
11      string citation("Les gens exigent la liberté d'expression pour compenser la liberté
        de pensée qu'ils préfèrent éviter.") ;
12      cout << "La citation de Soren Kirkegaard que vous préférez :" << endl ;
13      cout << citation << endl ;
14
15      return 0 ;
16  }
```

Je m'évalue

Exercice 3 (3 points)

1) Crée un nouveau fichier et nomme le **partie2.cpp**.

2) Complète la première colonne du tableau ci-dessous par les types de variables correspondant en C++.

Type	Nom	Valeur
	nbCochons	3
	masseJambon	25.5
	pigs	"petits cochons"
	nbLoup	1
	Reve	"Si j'avais rêvé de moutons électriques,"
	Cauchemar	"je serais mort électrocuté."

3) Utilise ces variables dans le programme **partie2.cpp** pour afficher le texte suivant sur la console de sortie :
Cette nuit, **1** loup a rêvé qu'il avait mangé **3 petits cochons**.
Ce matin, il a l'impression de peser **25.5** kg de plus.
Si j'avais rêvé de moutons électriques, se dit-il en se léchant les babines, **je serais mort électrocuté.**

Partie 3 : lecture de l'entrée

En C++, pour lire l'entrée au clavier d'une donnée, on utilise l'instruction "**cin**". C'est l'équivalent de l'instruction **print** en Python.
Pour placer cette donnée dans une variable, on utilise les chevrons ">>".

Attention, les chevrons ne sont plus dans le même sens que pour l'instruction "cout <<".
Crée un nouveau fichier et nomme-le **lecture.cpp**.
Tape le code ci-dessous dans ce fichier. Compile le puis exécute le. Étudie son fonctionnement.

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

int main() {

  string nom("") ; // déclaration de la variable nom et initialisation en chaîne vide
  cout << "Quel est ton nom ? " ; // affichage de la question
  cin >> nom ; // lecture de l'entrée

  // utilisation de l'entrée dans un affichage
  cout << "Bonjour " << nom << ". Seul un hominidé peut porter un tel nom." << endl ;

  return 0 ;
}
```

Je m'évalue

Exercice 4 (4 points)

1) Crée un nouveau fichier et nomme le **partie3.cpp**.

2) Ecris un programme en C++ dont quelques entrées-sorties sont décrites ci-dessous.

Exemples : **les réponses tapées par l'utilisateur sont en violet**
Entrée :
Quel est ton nom ? Bob
Quel âge as-tu ? 17
Quelle est ta taille (en m) ? 1.79
Sortie :
Bonjour Bob, tu mesures 179 cm et tu es né(e) en 2004 ou en 2005.

Entrée :
Quel est ton nom ? Alice
Quel âge as-tu ? 35
Quelle est ta taille (en m) ? 1.70
Sortie :
Bonjour Alice, tu mesures 170 cm et tu es né(e) en 1986 ou 1987.

Explications :
La taille est donnée en mètres par l'utilisateur et est affichée en centimètres dans le message de sortie. Elle devra donc être convertie. Pour convertir des centimètres en mètres, il faut multiplier par 100.
Pour les années de naissance, il faut calculer : 2022 – *dateNaissance* – 1 et 2022 – *dateNaissance*.

Partie 4 : boucles de répétitions

En C++, comme en Python, les boucles de répétitions se programment à l'aide de l'instruction **for**.
Cependant, la syntaxe n'est pas la même. En effet, en C++, la fonction **range( )** n'existe pas.
Voici la forme d'une boucle de répétition en C++ :

```
for (initialisation ; condition ; incrementation) {
  // instructions à répéter ;
}
```

1) Dans l'éditeur Spyder, écrire un mini-programme Python permettant d'afficher :
   0 1 2 3 4 5 6 7 8 9 10

2) Comment afficher cette même suite de nombres en C++ ?
   a) Partie initialisation.
   Il s'agit ici de déclarer une variable qu'on pourrait appeler **loop**.
   De manière analogue au langage Python, cette variable stockera les « numéros de boucles ».
   Comme ces numéros sont des nombres entiers (**int**) et que le numéro de départ est 0, il faudra donc déclarer puis initialiser notre variable **loop** :

```
for (int loop(0) ; condition ; incrementation) {
  // instructions à répéter ;
}
```

   b) Condition.
   Il s'agit ici de la condition d'arrêt. Quand l'instruction **for** doit-elle se terminer ?
   L'instruction **for** devra se terminer quand le numéro de la boucle sera égal à 11.
   Cette condition se traduit par :

```
for (int loop(0) ; loop < 11 ; incrementation) {
  // instructions à répéter ;
}
```

   c) Incrémentation.
   Il s'agit ici de déterminer le pas de la suite de nombres. Autrement dit, « de combien en combien faut-il compter » ? Ici, il faudra compter de 1 en 1. Il faudra donc ajouter 1 à la variable **loop** à chaque tour, ce qui se traduit par :

```
for (int loop(0) ; loop < 11 ; loop = loop + 1) {
  // instructions à répéter ;
}
```

   d) Affichage de la suite de nombres.
   On utilise l'instruction **cout <<** pour afficher les numéros de boucle successifs :

```
for (int loop(0) ; loop < 11 ; loop = loop + 1) {
  cout << loop << " " ;
}
```

3) Crée un nouveau fichier et nomme le **boucles.cpp**.
   Insère les lignes de codes de la question précédente dans un programme C++.
   Compile ce programme puis exécute le pour vérifier que la suite de nombres voulue s'affiche.

Je m'évalue

Exercice 5 (4 points)

1) Crée un nouveau fichier et nomme le **partie4.cpp**.

2) Utilise plusieurs instructions **for** pour qu'à l'exécution de ce programme, les suites de nombres ci-dessous s'affichent les unes sous les autres sur la console de sortie :

8 10 12 14 16 18 20
10 9 8 7 6 5 4 3 2 1 0
1 4 9 16 25 36 49 64 81 100

Explications :
La dernière suite de nombres est une suite de carrés :
1 × 1 = 1
2 × 2 = 4
3 × 3 = 9
Etc.

Partie 5 : les instructions conditionnelles

Les instructions **if** et **else** existent également en C++. L'instruction **elif** se note, en C++, **else if**.
Voici la forme de ces instructions :

```
// les conditions à tester doivent être placées entre parenthèses
if (condition1) {
    // instructions à exécuter si condition1 est vraie
}

else if (condition2) {
    // instructions à exécuter si condition2 est vraie
}

else {
    // instructions à exécuter
}
```

1) Crée un nouveau fichier et nomme le **age.cpp**.

2) En utilisant ces instructions, écris un programme en C++ qui demande son âge à l'utilisateur et affiche :

- "Vous êtes mineur(e)" si l'âge entré est inférieur ou égal à 17 ;
- "Vous êtes majeur(e)" si l'âge est compris entre 18 et 120 ;
- "Vous êtes meur(e) euh pardon, vous êtes mort(e)" si l'âge entré est strictement supérieur à 120.


Clique ici si tu ne vois pas du tout comment faire  **solution** 

Je m'évalue

Exercice 6 (4 points)

1) Crée un nouveau fichier et nomme le **partie5.cpp**.

2) Écris un programme en C++ qui demande à l'utilisateur le numéro du mois et qui renvoie le nombre de jours dans ce mois.
On assure que l'utilisateur entrera toujours un nombre entier compris entre 1 et 12 inclus.
On assure qu'il ne s'agit pas d'une année bissextile.

Exemples :

Entrée :	Entrée :	Entrée :	Entrée :	Entrée :
4	2	12	7	8
Sortie :	Sortie :	Sortie :	Sortie :	Sortie :
30	28	31	31	31

3) Écris la version Python de ce programme.

4) Quel est l'inconvénient dans l'écriture de ces programmes ?