

# INF1035 Informatique pour les sciences – Automne 2017

## Devoir 2 (TP2)

Date de remise : le 13 novembre à 23h00

(Source : Mohamed Bouguessa)

### Important

- 1) Ce TP est noté sur 10 points et il comporte trois exercices :
  - 1) Exercice 1 : 3 points
  - 2) Exercice 2 : 4 points
  - 3) Exercice 3 : 3 points
- 2) Le TP peut être fait en équipe de deux étudiants maximum.
- 3) Pour éviter les doublons, chaque équipe de deux étudiants doit remettre une seule fois le TP. Il faudra donc mentionner les noms des deux étudiants qui travaillent ensemble dans un fichier texte séparé (noms.txt).
- 4) Vous devez soumettre un seul fichier .zip qui contient tout les fichiers nécessaires pour tester votre travail.
  - a. Votre fichier zip doit être nommé comme suit :  
code\_permanent\_Etudiant1\_code\_permanent\_Etudiant2.zip
  - b. Le fichier compressé doit contenir 4 fichiers avec les noms suivants :
    - i. ex1.py (code pour l'exercice 1)
    - ii. ex2.py (code pour l'exercice 2)
    - iii. ex3.py (code pour l'exercice 3 )
    - iv. noms.txt (noms et codes permanents)
- 5) Vous devez soumettre votre travail via moodle seulement. **Les soumissions par courriel ne seront pas corrigées.**
- 6) La date de remise est lundi le 13 novembre à 23h00, **aucun TP ne sera accepté à partir de cette date.**
- 7) Le plagiat ne sera pas toléré, écrivez votre propre code. **Les normes de plagiat de l'université seront appliquées en cas de plagiat.**
- 8) Votre code ne sera pas traité s'il ne contient pas des commentaires clairs et précis.
- 9) N'oubliez pas de vous identifier. Indiquez votre nom et matricule dans chacun des fichiers que vous soumettez.
- 10) Afin d'être corrigés, vos programmes doivent fonctionner correctement sur les machines du laboratoire où les séances de laboratoire se déroulent.
- 11) Si vos programmes ne fonctionnent pas sur les machines désignées, alors les correcteurs n'auront pas l'obligation de corriger vos travaux.
- 12) Un travail non corrigé peut obtenir une note de zéro, et ce, sans possibilité de reprise. Il est de votre responsabilité de vous assurer que vos programmes sont compilables sur les machines disponibles au PK-S1565. Il est également de votre responsabilité de soumettre la **bonne** version de votre TP.
- 13) **RESPECTEZ TOUTES LES CONSIGNES.** Le correcteur pourra vous pénaliser dans le cas contraire.

# Énoncé

## Exercice 1 (3 points)

Cet exercice a pour but de programmer un questionnaire interactif dont le thème porte sur le réchauffement climatique. Votre tâche consiste à créer quatre questions à choix multiples. Pour chaque question, le programme propose 4 réponses possibles (numéroté de 1 à 4). Parmi ces quatre réponses, seulement une seule réponse est juste. Voici les quatre questions que vous devez considérer dans votre programme. La bonne réponse est illustrée (à titre indicatif seulement) en gras.

Combien de degré Celsius, en moyenne, la température a-t-elle augmenté depuis l'année 1800?

- 1: 0.4 degré Celsius
- 2: 0.6 degré Celsius**
- 3: 2.4 degré Celsius
- 4: 3.4 degré Celsius

On s'attend à ce qu'elle continue d'augmenter d'ici à l'an 2100. De combien de degrés ?

- 1 : De 0,5 à 3,6
- 2 : De 1,4 à 5,8**
- 3 : De 2,7 à 7,9
- 4 : De 0.6 a 2.3

Quel est le gaz à effet de serre le plus abondant dans l'atmosphère?

- 1 : La vapeur d'eau**
- 2 : Dioxyde de carbone
- 3 : Méthane
- 4 : Monoxyde de carbone

En quelle année est apparu le terme « effet de serre » ?

- 1 : 1824**
- 2 : 1896
- 3 : 1958
- 4 : 1962

Votre programme doit afficher une seule question à la fois ainsi que les quatre réponses possibles. Par la suite, l'utilisateur doit choisir une réponse. Si l'utilisateur fourni la bonne réponse, un message de félicitation s'affiche à l'écran. Sinon, un message d'erreur, qui informe l'utilisateur qu'il a fourni la mauvaise réponse, doit être affiché. Dans ce cas, l'utilisateur doit être également informé de la bonne réponse.

En terminant, si un utilisateur répond correctement à toutes les questions, votre programme doit afficher *Excellent*. Dans le cas de trois bonnes réponses, votre programme doit afficher *Très bien*; sinon (c.-à-d. deux réponses ou moins) un message qui invite l'utilisateur à améliorer ses connaissances s'affiche.

**Notes :**

- Vous devez commenter votre code (incluant les structures et les sélections)
- Utilisez les notions vues en classe. Ne répétez pas de code inutilement
- Pondération
  - Qualité du code (clarté, commentaires, variables bien définies, etc.) : **1.5 points**
  - Exécution du programme et exactitude des tests effectués : **1.5 points**

**Exercice 2 (4 points)**

Écrivez un script en Python qui permet de coder les règles de jeu suivantes :

Pour ce jeu on a besoin de deux dés de six faces. Chaque face contient les nombres 1, 2, 3, 4, 5 et 6.

Le joueur lance les deux dés.

Le joueur gagne, dès le premier tour, si la somme des deux faces de chacun des deux dés est égale à 7 ou à 11. Le jeu se termine.

Si la somme est égale à 2, 3 ou 12 (toujours dans le premier tour), le joueur perd. Le jeu se termine.  
Si la somme est égale à 4, 5, 6, 8, 9 ou bien 10 (toujours dans le premier tour), ce résultat détermine le « Point » et le deuxième tour du jeu commence.

(A) Le joueur relance les deux dés.

(B) Si la somme est égale à la valeur du « point » identifiée lors du premier tour, le joueur gagne.  
Le jeu se termine.

(C) Si la somme est égale à 7, le joueur perd. Le jeu se termine.

(D) Continuer de jouer, c.-à-d., refaire (A) et les tests (B) et (C).

Voici des exemples d'exécutions:

Exemple 1 :

```
Lancé 5 + 6 = 11  
Joueur gagne
```

Exemple 2 :

```
Lancé 4 + 6 = 10  
Le Point est 10  
Lancé 2 + 3 = 5  
Lancé 1 + 5 = 6  
Lancé 5 + 1 = 6  
Lancé 6 + 6 = 12  
Lancé 4 + 5 = 9  
Lancé 3 + 4 = 7  
Joueur perd
```

Exemple 3 :

```
Lancé 3 + 5 = 8  
Le Point est 8  
Lancé 1 + 5 = 6  
Lancé 4 + 3 = 7  
Joueur perd
```

Exemple 4 :

```
Lancé 6 + 3 = 9  
Le Point est 9  
Lancé 6 + 3 = 9  
Joueur gagne
```

Notes :

- Vous devez commenter votre code (incluant les structures et les sélections)
- Utilisez les notions vues en classe. Ne répétez pas de code inutilement
- Pondération
  - Qualité du code (clarté, commentaires, variables bien définies, etc.) : **2 points**
  - Exécution du programme et exactitude des tests effectués : **2 points**

### **Exercice 3 (3 points)**

Écrivez un programme qui, étant donné un entier impair, réalise le triangle suivant (si l'entier vaut 7 [pour 7 lignes]):

```
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

Voici des exemples d'exécutions :

**Exemple 1 :**

```
Donner un entier impair:3
1
1 2
1
```

**Exemple 2 :**

```
Donner un entier impair:5
1
1 2
1 2 3
1 2
1
```

### Exemple 3 :

```
Donner un entier impaire:7
1
1 2
1 2 3
1 2 3 4
1 2 3
1 2
1
```

### Exemple 4 :

```
Donner un entier impaire:9
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

### Notes :

- Vous devez commenter votre code (incluant les structures et les sélections)
- Pondération
  - Qualité du code (clarté, commentaires, variables bien définies, etc.) : **1 points**
  - Exécution du programme et exactitude des tests effectués : **2 points**