

# INF1035 Informatique pour les sciences – Automne 2017

## Devoir 3 (TP3)

Date de remise : le 15 décembre 2017 à 23h00

(Source : Mohamed Bouguessa)

### Important

- 1) Ce TP est noté sur 10 points et il comporte deux exercices :
  - a) Exercice 1 : 7 points
  - b) Exercice 2 : 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 tous 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 3 fichiers avec les noms suivants :
    - i. ex1.py (code pour l'exercice 1)
    - ii. ex2.py (code pour l'exercice 2)
    - iii. 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 vendredi 15 décembre 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 pouvoir bien fonctionner sur les machines 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 exécutables 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 (7 points)

Développer une application qui peut être utilisée par une salle de spectacle afin de vendre des billets. La capacité de cette salle, en terme de nombre de places, est de 450 sièges répartis sur 15 rangées à raison de 30 sièges par rangée.

Voici les tâches que vous devez implémenter :

Dès le lancement du programme, ce dernier demande à l'utilisateur de fournir le nom du fichier contenant les tarifs des sièges et ce pour chaque rangée. Vous pouvez choisir le format que vous voulez pour ce fichier d'input. Par exemple : si le tarif des sièges situés dans la rangée 1 est 200\$, le tarif des sièges situés dans la rangée 2 est 180\$, etc. alors les deux premières lignes du fichier pourraient être par exemple :

|        |
|--------|
| 1, 200 |
| 2,180  |

Une fois les prix des sièges (associés à chaque rangée) déterminés, votre programme affiche un menu de 5 options :

1. **Afficher les places disponibles**
2. **Afficher les tarifs**
3. **Afficher le total des ventes**
4. **Acheter un billet**
5. **Quitter**

**Veillez SVP choisir une option (1 - 5)**

- Si l'utilisateur choisit l'option 1 - Afficher les places disponibles : votre programme doit afficher la répartition des sièges tel qu'illustré par la figure suivante (les sièges vendus sont représentés par des \* alors que les sièges disponibles sont représentés par des #) :

```
Menu principal

1. Afficher les places disponibles
2. Afficher les tarifs
3. Afficher le total des ventes
4. Acheter un billet
5. Quitter

Veuillez SVP choisir une option (1 - 5) : 1

Sièges

Rangée 1      #####*#####
Rangée 2      *#*#####
Rangée 3      #####
Rangée 4      *****#####
Rangée 5      #####
Rangée 6      #####
Rangée 7      #####
Rangée 8      #####
Rangée 9      #####
Rangée 10     #####
Rangée 11     #####
Rangée 12     #####
Rangée 13     #####
Rangée 14     #####
Rangée 15     #####

Légende:      * = Vendu
              # = Disponible

Appuyez sur la touche Entrée pour continuer.█
```

Une fois l’affichage terminé, votre programme ne doit pas quitter mais il doit retourner au menu principal (le menu avec les 5 options).

- Si l'utilisateur choisit l'option 2 - Afficher les tarifs : votre programme doit afficher les tarifs associés à chaque rangée tel qu'illustré par la figure suivante :

| Veuillez SVP choisir une option (1 – 5) : 2  |        |
|--|--------|
| Prix des billets par rangée                  |        |
| Rangée                                       | Prix   |
| 1  | 200.00 |
| 2  | 180.00 |
| 3  | 160.00 |
| 4  | 140.00 |
| 5  | 120.00 |
| 6  | 100.00 |
| 7  | 80.00  |
| 8  | 60.00  |
| 9  | 40.00  |
| 10   | 30.00  |
| 11   | 30.00  |
| 12   | 30.00  |
| 13   | 30.00  |
| 14   | 30.00  |
| 15   | 30.00  |
| Appuyez sur la touche Entrée pour continuer. |        |

Une fois l’affichage des tarifs complété, votre programme ne doit pas quitter mais il doit retourner au menu principal (le menu avec les 5 options).

- Si l’utilisateur choisit l’option 3 - Afficher le total des ventes : votre programme doit afficher tout simplement le total des ventes enregistrées.

Une fois le total des ventes affiché, votre programme ne doit pas quitter mais il doit retourner au menu principal (le menu avec les 5 options).

- Si l’utilisateur choisit l’option 4 - Acheter un billet : d’abord un message s’affiche qui demande à l’utilisateur s’il veut voir ou non la figure qui montre les places disponibles.
  - Si l’utilisateur choisit oui, la figure s’affiche suivi d’un message qui demande de choisir la rangée ainsi que l’emplacement du siège qu’il veut acheter.
  - Si l’utilisateur choisit non, le message qui demande de choisir la rangée ainsi que l’emplacement du siège à acheter s’affiche directement.

Une fois le choix du siège rentré, votre programme doit faire une vérification pour voir si le siège est disponible ou non. Si le siège est disponible, la réservation est confirmée via un message. Sinon, le programme demande à l’utilisateur de choisir un autre siège.

Le programme demande encore une fois à l’utilisateur s’il veut acheter un autre billet. Si oui, le même processus (présenté dans le paragraphe précédent) est répété. Sinon, un message s’affiche indiquant le nombre de places réservées ainsi que le coût total pour cet achat. Voici un exemple :

Veuillez SVP choisir une option (1 - 5) : 4  
Achat de billets

Voulez vous visualiser les places disponibles?  
avant de faire vos sélections (o/n)? o

Sièges

|        |    |       |
|--------|----|-------|
| Rangée | 1  | ##### |
| Rangée | 2  | ##### |
| Rangée | 3  | ##### |
| Rangée | 4  | ##### |
| Rangée | 5  | ##### |
| Rangée | 6  | ##### |
| Rangée | 7  | ##### |
| Rangée | 8  | ##### |
| Rangée | 9  | ##### |
| Rangée | 10 | ##### |
| Rangée | 11 | ##### |
| Rangée | 12 | ##### |
| Rangée | 13 | ##### |
| Rangée | 14 | ##### |
| Rangée | 15 | ##### |

Légende:        \* = Vendu  
              # = Disponible

Appuyez sur la touche Entrée pour continuer. █

SVP entrer le numéro de la rangée (1-15): 1

SVP entrer le numéro du siège (1-30): 15

Achat confirmé

Voulez-vous acheter un autre siège (o/n)? o

SVP entrer le numéro de la rangée (1-15): 1

SVP entrer le numéro du siège (1-30): 15

Désolé. Ce siège a été vendu.

Voulez-vous acheter un autre siège (o/n)? o

SVP entrer le numéro de la rangée (1-15): 1

SVP entrer le numéro du siège (1-30): 16

Achat confirmé

Voulez-vous acheter un autre siège (o/n)? n

Vous avez acheté un total de 2 billets pour un prix total de \$400.00

Menu principal

1. Afficher les places disponibles
2. Afficher les tarifs
3. Afficher le total des ventes
4. Acheter un billet
5. Quitter

Veuillez SVP choisir une option (1 - 5) : █

Prendre note que à chaque fois qu'une place est réservée, vous devez mettre à jour (en mémoire) la table qui contient les informations sur les places disponibles et réservées (il faut remplacer tout simplement # par \* lorsqu'une place spécifique est prise).

Une fois l'opération de vente complétée, votre programme ne doit pas quitter mais il doit retourner au menu principal (le menu avec les 5 options).

- Si l'utilisateur choisit l'option 5 - `Quitter` : votre programme doit terminer en affichant le nombre total de places vendues et le montant total des ventes.

#### Notes :

- Vous devez commenter votre code (incluant les structures et les sélections)
- Utilisez les notions vues en classe (liste, fonctions, etc.). Ne répétez pas de code inutilement
- Pondération
  - Qualité du code (clarté, commentaires, variables bien définies, etc.) : **3 points**
  - Exécution du programme et exactitude des tests effectués : **4 points**

### Exercice 2 (3 points)

Écrivez un programme qui simule le jeu Tic-Tac-Toe.

- Votre programme doit utiliser une liste à deux dimensions pour simuler le jeu en mémoire.
- Au lancement, votre programme doit sauvegarder des nombres aléatoires, zéro ou un, dans chaque élément de la liste. Le nombre zéro représente la lettre O, alors que le nombre un représente la lettre X.
- Une fois la liste remplie, votre programme doit afficher les lettres (O et X) de telle sorte que chaque case corresponde à un élément de la liste. Il s'agit d'un affichage simple, seulement les caractères et non un graphique comme dans les exemples. Par exemple :

|     |
|-----|
| OXO |
| XXO |
| OOX |

- Votre programme doit ensuite afficher un message pour indiquer si
  - Le joueur X a gagné
  - Ou bien le joueur Y a gagné
  - Ou bien aucun gagnant

Voici quelques exemples :

**Exemple 1 : aucun gagnant**

|   |   |   |
|---|---|---|
| O | X | O |
| X | X | O |
| O | O | X |

**Exemple 2 : le joueur O gagne**

|   |   |   |
|---|---|---|
| X | X | O |
| X | O | X |
| O | X | O |

**Exemple 3 : le joueur X gagne**

|   |   |   |
|---|---|---|
| X | O | O |
| O | X | O |
| O | X | X |

**Notes :**

- Vous devez commenter votre code (incluant les structures et les sélections)
- Utilisez les notions vues en classe (liste, fonctions, etc.). Ne répétez pas de code inutilement
- Pondération
  - Qualité du code (clarté, commentaires, variables bien définies, etc.) : **1 points**
  - Exécution du programme et exactitude des tests effectués : **2 points**