

Introduction à l'informatique

L1 MI/LDD1 IM

Nom:

Prénom:

Groupe:

TP Noté : sujet A

durée 1h30 (tiers temps additionnel 30 minutes)

Modalités

Le TP doit être rendu dans les cinq minutes de la fin du temps imparti :

- la feuille contenant le QCM est à rendre au chargé de TP
- le fichier Python à compléter est disponible sur la page du cours :

https://usr.lmf.cnrs.fr/~kn/ii_fr.html

et doit être déposé sur l'espace eCampus du cours **Intro Info**

L'accès à Internet n'est pas bloqué dans la salle de TP. Vous pouvez donc accéder aux documents tels que les corrigés de TP ou les supports de cours. Il est cependant de votre intérêt de ne pas passer trop de temps à chercher des réponses que vous êtes censés connaître. Les téléphones portables doivent être éteints. Toute communication entre étudiants est interdite. De même, il est interdit d'accéder à votre messagerie, *drive*, etc. L'accès à des sites autres que celui du cours, en particulier aux sites d'IA générative est **interdit**. En cas de doute, le surveillant de la salle peut inspecter votre session et regarder quelles pages et applications sont ouvertes. En cas de **suspicion**, un PV de fraude sera établi. La section disciplinaire de l'Université décidera ensuite du caractère avéré ou non de la fraude.

1 QCM de cours (5 points)

Pour chacune des questions, entourer la réponse correcte (vrai ou faux) :

- une bonne réponse donne 0,5 point
- une mauvaise réponse donne -0,5 point
- une absence de réponse donne 0 point

La note minimale pour une question est de 0 point (pas de score négatif).

1. Sur un système Unix ...

- | | | |
|---|------|------|
| (a) La commande <code>ls</code> permet de lister des fichiers | vrai | faux |
| (b) Le motif <i>glob</i> « <code>[a-z]*</code> » représente tous les noms de fichiers constitués uniquement de lettres minuscules | vrai | faux |
| (c) Un fichier avec les permissions 411 est modifiable par son propriétaire | vrai | faux |

2. En Python ...

- | | | |
|---|------|------|
| (a) Les caractères d'une chaîne sont modifiables (i.e. on peut écrire <code>s[i] = 'a'</code>) | vrai | faux |
| (b) L'expression <code>len([4,5,6])</code> renvoie 3 | vrai | faux |
| (c) Les entiers sont de taille fixe (32 ou 64 bits selon la machine) | vrai | faux |

3. En Python ...

- | | | |
|--|------|------|
| (a) On peut modifier une variable globale depuis une fonction | vrai | faux |
| (b) Il est obligatoire de mettre une instruction <code>return</code> dans une fonction | vrai | faux |
| (c) Si une fonction <code>f</code> contient l'expression <code>1/"toto"</code> et que cette fonction n'est pas appelée, l'interprète signale quand même une erreur dans <code>f</code> | vrai | faux |
| (d) <code>ord("é")</code> renvoie une valeur strictement inférieure à 128 | vrai | faux |

2 Programmation Python (15 points)

Télécharger le fichier Python **correspondant au sujet A** sur la page du cours. Renseignez en début de fichier votre nom, prénom, adresse email (@universite-paris-saclay.fr). Complétez le code de chacune des fonctions demandées. Chaque fonction ne contient qu'une instruction **return** que **vous devez remplacer par votre code**. Vous **ne devez pas modifier les commentaires** de la forme **#QUESTION 1** ou **#FIN QUESTION 1** se trouvant dans le fichier. Tout le code nécessaire à une question (par exemple, si vous décidez d'ajouter une fonction auxiliaire) doit se trouver entre les deux commentaires de début et de fin de la question. Par contre, tout code de test que vous écrirez doit se trouver en dehors de ces commentaires (par exemple en fin de fichier après le commentaire **#FIN QUESTION 7**).

1. (1,5 point) Compléter la fonction **conversion(n)** qui convertit un nombre **n** de secondes en minutes et en secondes. Si **n** est strictement négatif, la fonction renvoie la chaîne **"invalid"**. Sinon, elle renvoie la chaîne **"XminYs"** où **X** est le nombre de minutes et **Y** est le nombre de secondes.
2. (1,5 point) Compléter la fonction **test_q1()** pour qu'elle effectue deux tests au moyen de l'instruction **assert** :
 - 101 secondes font 1 minute 41 secondes
 - un test où la fonction renvoie la chaîne **"invalid"**
3. (2,5 points) Compléter la fonction **verif_tab(tab)** pour qu'elle renvoie **True** si le tableau **tab** est de taille paire et si les valeurs du tableau sont strictement croissantes (en particulier la fonction renvoie **True** pour le tableau vide). Dans tous les autres cas, la fonction renvoie **False**.
4. (1,5 point) Compléter la fonction **test_q3()** pour qu'elle teste au moyen d'**assert** la fonction **verif_tab** sur deux tableaux de taille paire (un correct, et un incorrect) et sur un tableau de taille impaire. Les trois tableaux doivent être de taille supérieure à trois (la fonction contient 3 instructions **assert**, pas une seule qui teste les trois tableaux).
5. (3 points) Compléter la fonction **verif_matrice(m, k)**. Cette dernière prend en argument un tableau de tableaux d'entiers et un entier **k** et renvoie
 - **True** si le tableau **m** est une matrice carrée (tous les tableaux internes ont la même taille que le tableau externe) et que la somme de tous les entiers du tableau est inférieure ou égale à **k**
 - **False** sinon

Par exemple si **m** vaut :

```
[[0, 1, 4],  
 [3, -4, 2],  
 [5, 6, 2]]
```

alors **verif_matrice(m, 20)** renvoie **True**.

6. (2 points) Compléter la fonction **test_q5()** pour qu'elle effectue trois tests au moyen d'**assert**. Un test doit être sur une matrice de taille 4x4, contenant des valeurs non toutes identiques et pour laquelle **verif_matrice(m, 20)** doit renvoyer **True**. Les deux autres **assert** doivent tester que **verif_matrice** renvoie bien **False** quand le tableau d'entrée n'est pas une matrice carrée ou quand l'une des sommes d'une ligne est plus grande que l'argument **k**.
7. (3 points) Compléter la fonction **recherche(tab)**. Cette dernière prend en argument un tableau de chaînes de caractères. Chaque chaîne est de la forme « **"Prénom:10"** » et représente le prénom d'un étudiant suivi de : suivi d'une note (un entier positif), sans espace. Si le tableau d'entrée est vide, votre fonction doit renvoyer la chaîne vide. Sinon votre fonction doit renvoyer le prénom associé à la note la plus haute. Si plusieurs lignes ont la même note maximale, vous pouvez renvoyer n'importe lequel des prénoms correspondants.

Indications : on rappelle qu'étant donné une chaîne **s**, **s.split(":")** permet d'obtenir un tableau de deux chaînes (le prénom et les notes). Un tableau de test **TAB_TEST** est fourni, votre fonction doit renvoyer **"Alice"** (car cette ligne contient 19, la meilleure note globale).

Introduction à l'informatique

L1 MI/LDD1 IM

Nom:

Prénom:

Groupe:

TP Noté : sujet B

durée 1h30 (tiers temps additionnel 30 minutes)

Modalités

Le TP doit être rendu dans les cinq minutes de la fin du temps imparti :

- la feuille contenant le QCM est à rendre au chargé de TP
- le fichier Python à compléter est disponible sur la page du cours :

https://usr.lmf.cnrs.fr/~kn/ii_fr.html

et doit être déposé sur l'espace eCampus du cours **Intro Info**

L'accès à Internet n'est pas bloqué dans la salle de TP. Vous pouvez donc accéder aux documents tels que les corrigés de TP ou les supports de cours. Il est cependant de votre intérêt de ne pas passer trop de temps à chercher des réponses que vous êtes censés connaître. Les téléphones portables doivent être éteints. Toute communication entre étudiants est interdite. De même, il est interdit d'accéder à votre messagerie, *drive*, etc. L'accès à des sites autres que celui du cours, en particulier aux sites d'IA générative est **interdit**. En cas de doute, le surveillant de la salle peut inspecter votre session et regarder quelles pages et applications sont ouvertes. En cas de **suspicion**, un PV de fraude sera établi. La section disciplinaire de l'Université décidera ensuite du caractère avéré ou non de la fraude.

1 QCM de cours (5 points)

Pour chacune des questions, entourer la réponse correcte (vrai ou faux) :

- une bonne réponse donne 0,5 point
- une mauvaise réponse donne -0,5 point
- une absence de réponse donne 0 point

La note minimale pour une question est de 0 point (pas de score négatif).

1. Sur un système Unix ...

- | | | |
|--|------|------|
| (a) La commande mkdir permet de lister des fichiers | vrai | faux |
| (b) Le motif <i>glob</i> « *[A-Z]* » représente tous les noms de fichiers ne contenant pas de chiffre | vrai | faux |
| (c) Un fichier avec les permissions 644 est exécutable par son propriétaire | vrai | faux |

2. En Python ...

- | | | |
|---|------|------|
| (a) Les cases d'un tableau sont modifiables (i.e. on peut écrire t[i] = 42) | vrai | faux |
| (b) L'expression len ([[3,4,5]]) renvoie 3 | vrai | faux |
| (c) Tout nombre réel peut être représenté par un nombre flottant de Python | vrai | faux |

3. En Python ...

- | | | |
|---|------|------|
| (a) On peut accéder à une variable locale à l'extérieur de la fonction où elle est définie | vrai | faux |
| (b) L'expression "2" > "10" vaut True | vrai | faux |
| (c) On peut utiliser l'instruction return en dehors d'une fonction | vrai | faux |
| (d) On peut définir dans une fonction une variable locale ayant le même nom qu'une variable globale | vrai | faux |

2 Programmation Python (15 points)

Télécharger le fichier Python **correspondant au sujet B** sur la page du cours. Renseignez en début de fichier votre nom, prénom, adresse email (@univ~~ers~~ite-paris-saclay.fr). Complétez le code de chacune des fonctions demandées. Chaque fonction ne contient qu'une instruction **return** que **vous devez remplacer par votre code**. Vous **ne devez pas modifier les commentaires** de la forme **#QUESTION 1** ou **#FIN QUESTION 1** se trouvant dans le fichier. Tout le code nécessaire à une question (par exemple, si vous décidez d'ajouter une fonction auxiliaire) doit se trouver entre les deux commentaires de début et de fin de la question. Par contre, tout code de test que vous écrirez doit se trouver en dehors de ces commentaires (par exemple en fin de fichier après le commentaire **#FIN QUESTION 7**).

- (1,5 point) Compléter la fonction **conversion(n)** qui convertit un nombre **n** représentant des degrés Celsius en degrés Fahrenheit. Si **n** est strictement inférieur à 273, la fonction renvoie la chaîne **"invalide"** Sinon, elle renvoie la chaîne **"X F"** où **X** est le nombre de degré Fahrenheit. La formule de conversion est $\frac{n \times 9}{5} + 32$. **On utilisera la division entière.**
- (1,5 point) Compléter la fonction **test_q1()** pour qu'elle effectue deux tests au moyen de l'instruction **assert** :
 - pour l'argument 100, la fonction renvoie la chaîne **"100 F"**
 - pour l'argument -300, la fonction renvoie la chaîne **"invalide"**
- (2,5 points) Compléter la fonction **verif_tab(tab)** pour qu'elle renvoie **True** si le tableau **tab** est de taille impaire et si chaque entier est le double de celui se trouvant dans la case précédente (la valeur de la case d'indice 0 peut être quelconque). Dans tous les autres cas, la fonction renvoie **False**.
- (1,5 point) Compléter la fonction **test_q3()** pour qu'elle teste au moyen d'**assert** la fonction **alt_tab** sur deux tableaux de taille impaire (un correct, et un incorrect) et sur un tableau de taille paire. Les trois tableaux doivent être de taille supérieure à trois (la fonction contient 3 instructions **assert**, pas une seule qui teste les trois tableaux).
- (3 points) Compléter la fonction **verif_matrice(m, k)**. Cette dernière prend en argument un tableau de tableaux d'entiers et deux entiers **a** et **k** et renvoie
 - **True** si le tableau **m** est une matrice carrée (tous les tableaux internes ont la même taille que le tableau externe) et il y a au moins **k** entiers dont la valeur est strictement inférieure à 10.
 - **False** sinonPar exemple si **m** vaut :

```
[[0, 1, 40],  
 [31, -4, 2],  
 [5, 72, 2]]
```

alors **verif_matrice(m, 5)** renvoie **True**.

- (2 points) Compléter la fonction **test_q5()** pour qu'elle effectue trois tests au moyen d'**assert**. Un test doit être sur une matrice de taille 4x4, contenant des valeurs non toutes identiques et pour laquelle **verif_matrice(m, 5)** doit renvoyer **True**. Les deux autres **assert** doivent tester que **verif_matrice** renvoie bien **False** quand le tableau d'entrée n'est pas une matrice carrée ou quand il n'y a pas au moins **k** entiers inférieurs à 10.
- (3 points) Compléter la fonction **moyenne(tab)**. Cette dernière prend en argument un tableau de chaînes de caractères. Chaque chaîne est de la forme « **"Ville,30"** » et représente une ville suivie de la température dans cette ville (un entier, séparé du nom de ville par une virgule sans espace). Si le tableau d'entrée est vide, votre fonction doit renvoyer le flottant **0.0**. Sinon votre fonction doit renvoyer la moyenne des températures, sous la forme d'un nombre flottant.
Indications : on rappelle qu'étant donné une chaîne **s** au format indiqué, **s.split(",")** permet d'obtenir un tableau contenant deux **chaînes** (ville, température maximale). Un tableau de test **TAB_TEST** est fourni, votre fonction doit renvoyer **33.5** pour ce tableau.

Introduction à l'informatique

L1 MI/LDD1 IM

Nom:

Prénom:

Groupe:

TP Noté : sujet C

durée 1h30 (tiers temps additionnel 30 minutes)

Modalités

Le TP doit être rendu dans les cinq minutes de la fin du temps imparti :

- la feuille contenant le QCM est à rendre au chargé de TP
- le fichier Python à compléter est disponible sur la page du cours :

https://usr.lmf.cnrs.fr/~kn/ii_fr.html

et doit être déposé sur l'espace eCampus du cours **Intro Info**

L'accès à Internet n'est pas bloqué dans la salle de TP. Vous pouvez donc accéder aux documents tels que les corrigés de TP ou les supports de cours. Il est cependant de votre intérêt de ne pas passer trop de temps à chercher des réponses que vous êtes censés connaître. Les téléphones portables doivent être éteints. Toute communication entre étudiants est interdite. De même, il est interdit d'accéder à votre messagerie, *drive*, etc. L'accès à des sites autres que celui du cours, en particulier aux sites d'IA générative est **interdit**. En cas de doute, le surveillant de la salle peut inspecter votre session et regarder quelles pages et applications sont ouvertes. En cas de **suspicion**, un PV de fraude sera établi. La section disciplinaire de l'Université décidera ensuite du caractère avéré ou non de la fraude.

1 QCM de cours (5 points)

Pour chacune des questions, entourer la réponse correcte (vrai ou faux) :

- une bonne réponse donne 0,5 point
- une mauvaise réponse donne -0,5 point
- une absence de réponse donne 0 point

La note minimale pour une question est de 0 point (pas de score négatif).

1. Sur un système Unix ...

- | | | |
|--|------|------|
| (a) La commande cd permet de lister des fichiers | vrai | faux |
| (b) Le motif <i>glob</i> « *[0-9] » représente tous les noms de fichiers terminant par un chiffre | vrai | faux |
| (c) Le propriétaire d'un répertoire avec les permissions 600 peut en faire le répertoire courant. | vrai | faux |

2. En Python ...

- | | | |
|--|------|------|
| (a) L'expression "1" + "2" ne provoque pas d'erreur | vrai | faux |
| (b) L'expression len (3 * [4]) renvoie 3 | vrai | faux |
| (c) L'expression int ("foo") renvoie 0 | vrai | faux |

3. En Python ...

- | | | |
|---|------|------|
| (a) On peut accéder à une même variable globale depuis deux fonctions différentes | vrai | faux |
| (b) La fonction range peut prendre entre 1 et 3 arguments | vrai | faux |
| (c) Un bloc try ... except: ... permet de rattraper une erreur (exception) | vrai | faux |
| (d) On peut utiliser le contenu d'une variable avant de l'avoir initialisée | vrai | faux |

2 Programmation Python (15 points)

Télécharger le fichier Python **correspondant au sujet C** sur la page du cours. Renseignez en début de fichier votre nom, prénom, adresse email (@**universite-paris-saclay.fr**). Complétez le code de chacune des fonctions demandées. Chaque fonction ne contient qu'une instruction **return** que **vous devez remplacer par votre code**. Vous **ne devez pas modifier les commentaires** de la forme **#QUESTION 1** ou **#FIN QUESTION 1** se trouvant dans le fichier. Tout le code nécessaire à une question (par exemple, si vous décidez d'ajouter une fonction auxiliaire) doit se trouver entre les deux commentaires de début et de fin de la question. Par contre, tout code de test que vous écrirez doit se trouver en dehors de ces commentaires (par exemple en fin de fichier après le commentaire **#FIN QUESTION 7**).

- (1,5 point) Compléter la fonction **conversion(n)** qui convertit une heure (comprise entre 0 et 23) en notation anglo-saxonne : les heures 0, 1, 2, ..., 11, valent "12AM", "1AM", "2AM", ..., "11AM". Les heures entre 12 et 23 valent "12PM", "1PM", ..., "11PM". Le résultat est renvoyé sous forme d'une chaîne de caractères. Si **n** n'est pas entre 0 et 23, la fonction renvoie la chaîne "invalid".
- (1,5 point) Compléter la fonction **test_q1()** pour qu'elle effectue deux tests au moyen de l'instruction **assert** :
 - pour 13 votre fonction renvoie "1PM"
 - pour -10 votre fonction renvoie "invalid"
- (2,5 points) Compléter la fonction **verif_tab(tab)** pour qu'elle renvoie **True** si le tableau **tab** est de taille paire et ne contient que des entiers positifs ou nuls aux indices pairs et des entiers strictement négatifs aux indices impairs. Si l'une de ces conditions n'est pas remplie, la fonction renvoie **False**.
- (1,5 point) Compléter la fonction **test_q3()** pour qu'elle teste au moyen d'**assert** la fonction **verif_tab** sur deux tableaux de taille paire (un correct, et un incorrect) et pour un tableau de taille impaire. Les trois tableaux doivent être de taille supérieure à trois (la fonction contient 3 instructions **assert**, pas une seul qui teste les trois tableaux).
- (3 points) Compléter la fonction **verif_matrix(m, k)**. Cette dernière prend en argument un tableau de tableaux d'entiers et un entier **k** et renvoie
 - **True** si le tableau **m** est une matrice carrée (tous les tableaux internes ont la même taille que le tableau externe) et si la somme de chaque ligne est inférieure à **k**
 - **False** sinonPar exemple si **m** vaut :

```
[[1, 1, 4],
 [3, -4, -2],
 [5, 7, 2]]
```

alors **verif_matrix(m, 20)** renvoie **True**.

- (2 points) Compléter la fonction **test_q5()** pour qu'elle effectue trois tests au moyen d'**assert**. Un test doit être sur une matrice de taille 4x4, contenant des valeurs non toutes identiques et pour laquelle **verif_matrix(m, 20)** doit renvoyer **True**. Les deux autres **assert** doivent tester que **verif_matrix** renvoie bien **False** quand le tableau d'entrée n'est pas une matrice carrée ou quand la somme d'une ligne est strictement supérieure à **k**.
- (3 points) Compléter la fonction **max_inter(tab)**. Cette dernière prend en argument un tableau de chaînes de caractères. Chaque chaîne est de la forme « "-3:30" » et représente un intervalle entre deux entiers, le deuxième entier toujours supérieur au premier (vous n'avez pas à le vérifier). Si le tableau d'entrée est vide, votre fonction doit renvoyer 0. Sinon votre fonction doit renvoyer la taille de l'intervalle le plus grand (la taille d'un intervalle est la différence entre sa borne supérieure et sa borne inférieure)

Indications : on rappelle qu'étant donné une chaîne **s** au format indiqué, **s.split(":")** permet d'obtenir un tableau de deux chaînes (borne inférieure, borne supérieure). Un tableau de test **TAB_TEST** est fourni, votre fonction doit renvoyer **100** pour ce tableau.

Introduction à l'informatique

L1 MI/LDD1 IM

Nom:

Prénom:

Groupe:

TP Noté : sujet D

durée 1h30 (tiers temps additionnel 30 minutes)

Modalités

Le TP doit être rendu dans les cinq minutes de la fin du temps imparti :

- la feuille contenant le QCM est à rendre au chargé de TP
- le fichier Python à compléter est disponible sur la page du cours :

https://usr.lmf.cnrs.fr/~kn/ii_fr.html

et doit être déposé sur l'espace eCampus du cours **Intro Info**

L'accès à Internet n'est pas bloqué dans la salle de TP. Vous pouvez donc accéder aux documents tels que les corrigés de TP ou les supports de cours. Il est cependant de votre intérêt de ne pas passer trop de temps à chercher des réponses que vous êtes censés connaître. Les téléphones portables doivent être éteints. Toute communication entre étudiants est interdite. De même, il est interdit d'accéder à votre messagerie, *drive*, etc. L'accès à des sites autres que celui du cours, en particulier aux sites d'IA générative est **interdit**. En cas de doute, le surveillant de la salle peut inspecter votre session et regarder quelles pages et applications sont ouvertes. En cas de **suspicion**, un PV de fraude sera établi. La section disciplinaire de l'Université décidera ensuite du caractère avéré ou non de la fraude.

1 QCM de cours (5 points)

Pour chacune des questions, entourer la réponse correcte (vrai ou faux) :

- une bonne réponse donne 0,5 point
- une mauvaise réponse donne -0,5 point
- une absence de réponse donne 0 point

La note minimale pour une question est de 0 point (pas de score négatif).

1. Sur un système Unix ...

- | | | |
|---|------|------|
| (a) La commande cp permet de déplacer des fichiers | vrai | faux |
| (b) Le motif <i>glob</i> « [0-9]*[0-9] » représente tous les noms de fichiers ne contenant pas de lettre | vrai | faux |
| (c) Un fichier dont les permissions sont 400 peut être modifié par son propriétaire | vrai | faux |

2. En Python ...

- | | | |
|---|------|------|
| (a) Si t = [1, 2], alors t[2] provoque une erreur | vrai | faux |
| (b) L'expression 3 * [4] renvoie 12 | vrai | faux |
| (c) L'expression int("123") renvoie 123 | vrai | faux |

3. En Python ...

- | | | |
|--|------|------|
| (a) Une fonction peut lire une variable globale X sans la mentionner dans une instruction global X | vrai | faux |
| (b) On doit obligatoirement mettre un bloc else: après un bloc if cond: | vrai | faux |
| (c) Mettre à jour la case d'un tableau peut modifier la longueur du tableau | vrai | faux |
| (d) "A" * str(3) provoque une erreur | vrai | faux |

2 Programmation Python (15 points)

Télécharger le fichier Python **correspondant au sujet D** sur la page du cours. Renseignez en début de fichier votre nom, prénom, adresse email (@univ~~ers~~ite-paris-saclay.fr). Complétez le code de chacune des fonctions demandées. Chaque fonction ne contient qu'une instruction **return** que **vous devez remplacer par votre code**. Vous **ne devez pas modifier les commentaires** de la forme **#QUESTION 1** ou **#FIN QUESTION 1** se trouvant dans le fichier. Tout le code nécessaire à une question (par exemple, si vous décidez d'ajouter une fonction auxiliaire) doit se trouver entre les deux commentaires de début et de fin de la question. Par contre, tout code de test que vous écrirez doit se trouver en dehors de ces commentaires (par exemple en fin de fichier après le commentaire **#FIN QUESTION 7**).

- (1,5 point) Compléter la fonction **conversion(n)** qui convertit un nombre **n** de bits en octets et en bits. Si **n** est strictement négatif, la fonction renvoie la chaîne **"invalid"**. Sinon, elle renvoie la chaîne **"XoYb"** où **X** est le nombre d'octets et **Y** est le nombre de bits.
- (1,5 point) Compléter la fonction **test_q1()** pour qu'elle effectue deux tests au moyen de l'instruction **assert** :
 - pour 107 bits la fonction renvoie **"13o3b"**
 - pour un nombre négatif, la fonction renvoie **"invalid"**
- (2,5 points) Compléter la fonction **verif_(tab)** pour qu'elle renvoie **True** si le tableau **tab** est de taille impaire et si chaque case diffère de 2 par rapport à la précédente (la fonction valeur absolue est **abs**). Si l'une de ces conditions n'est pas remplie, la fonction renvoie **False**.
- (1,5 point) Compléter la fonction **test_q3()** pour qu'elle teste au moyen d'**assert** la fonction **verif_tab** sur deux tableaux de taille impaire (un correct, et un incorrect) et pour un tableau de taille paire. Les trois tableaux doivent être de taille supérieure ou égale à trois (la fonction contient 3 instructions **assert**, pas une seule qui teste les trois tableaux).
- (3 points) Compléter la fonction **verif_matrice(m, k)**. Cette dernière prend en argument un tableau de tableaux d'entiers et un nombre flottant **k** et renvoie
 - **True** si le tableau est une matrice carrée (tous les tableaux internes ont la même taille que le tableau externe) et la moyenne de toutes les valeurs du tableau est supérieure ou égale à **k** (votre fonction renvoie **True** pour le tableau vide, pour toute valeur de **k**).
 - **False** sinon

Par exemple si **m** vaut :

```
[[1, 1, 4],
 [3, 4, 2],
 [5, 7, 2]]
```

alors **verif_matrice(m, 1.5)** renvoie **True**.

- (2 points) Compléter la fonction **test_q5()** pour qu'elle effectue trois tests au moyen d'**assert**. Un test doit être sur une matrice de taille 4x4, contenant des valeurs non toutes identiques et pour laquelle **verif_matrice(m, 5)** doit renvoyer **True**. Les deux autres **assert** doivent tester que **verif_matrice** renvoie bien **False** quand le tableau d'entrée n'est pas une matrice carrée ou quand l'une des moyenne est strictement inférieure à **k**.
- (3 points) Compléter la fonction **max_amount(tab)**. Cette dernière prend en argument un tableau de chaînes de caractères. Chaque chaîne est de la forme « **"100,retrait"** » et représente une somme en euros (entier positif ou négatif) et un intitulé d'opération bancaire. Si le tableau d'entrée est vide, votre fonction doit renvoyer la chaîne de caractères vide. Sinon votre fonction doit renvoyer l'intitulé de l'opération ayant la plus grande valeur absolue.

Indications : on rappelle qu'étant donné une chaîne **s** au format indiqué, **s.split(",")** permet d'obtenir un tableau de deux **chaînes** (montant, intitulé). Un tableau de test **TAB_TEST** est fourni, votre fonction doit renvoyer **"salaire"**.