

## Sujet 1 : niveau moyen

### Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

### Exercice 1 *Fonction et conditionnelles*

1. Écrire une fonction `max2(a, b)` qui renvoie le maximum des arguments entiers `a` et `b` sans utiliser la fonction `built-in max`.

Mettre entre triples guillemets, les tests effectués pour vérifier la correction de cette fonction.

```
>>> max2(10, 8)
10
```

2. Écrire une fonction `max3(a, b, c)` qui renvoie le maximum des arguments entiers `a` et `b` sans utiliser la fonction `built-in max`.

Mettre entre triples guillemets, les tests effectués pour vérifier la correction de cette fonction.

```
>>> max3(10, 8, 10)
10
```

### Exercice 2 *Boucle interactive*

Écrire un programme qui demande d'abord à l'utilisateur un nombre entier de notes tant que le nombre saisi n'est pas strictement positif.

Ensuite le programme demande à l'utilisateur de saisir chaque note et affiche en sortie la note maximale et le nombre de fois où elle est atteinte.



Interdiction d'utiliser les listes/tableaux Python.

Voici une trace d'exécution du programme :

```
Nombre de notes ? -2
Saisir un nombre > 0, nombre de notes ? -1
Saisir un nombre > 0, nombre de notes ? 3
Nouvelle note ? 10
Nouvelle note ? 8
Nouvelle note ? 10
Note maximale : 10.0 atteinte 2 fois
```

## Sujet 2 : niveau moyen

### Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

### Exercice 1 *Fonction et conditionnelles*

1. Écrire une fonction `bac(moyenne)` qui prend en argument une moyenne de type `float` entre 0 et 20, vérifie si la moyenne est entre 0 et 20 avec une instruction `assert` et renvoie :
  - `True` si `moyenne >= 10`
  - `False` si `moyenne < 10`
2. Écrire une fonction `mention(moyenne)` qui prend en argument une moyenne de type `float` entre 0 et 20 et qui renvoie une chaîne de caractère :
  - `"recalé"` si  $0 \leq \text{note} < 8$
  - `"second groupe"` si  $8 \leq \text{note} < 10$
  - `"reçu"` si  $10 \leq \text{note} < 12$
  - `"assez bien"` si  $12 \leq \text{note} < 14$
  - `"bien"` si  $14 \leq \text{note} < 16$
  - `"très bien"` si  $16 \leq \text{note} \leq 20$
  - `"valeur incohérente"` sinon

On donne quelques tests unitaires qui doivent être vérifiés. Compléter ces tests en essayant de couvrir tous les cas possibles. Mettre entre triples guillemets, les tests effectués.

```
>>> assert mention(9) == "second groupe"  
>>> assert mention(-1) == "valeur incohérente"
```

## Exercice 2 *Boucle interactive*

Écrire un programme qui demande d'abord à l'utilisateur un nombre entier de notes tant que le nombre saisi n'est pas strictement positif.

Ensuite le programme demande à l'utilisateur de saisir chaque note et affiche en sortie la moyenne des notes saisies, arrondie au centième. On utilisera la fonction `round`, pour afficher sa documentation, saisir `help(round)` dans une console interactive.

Attention! Interdiction d'utiliser les listes/tableaux Python.

Voici une trace d'exécution du programme :

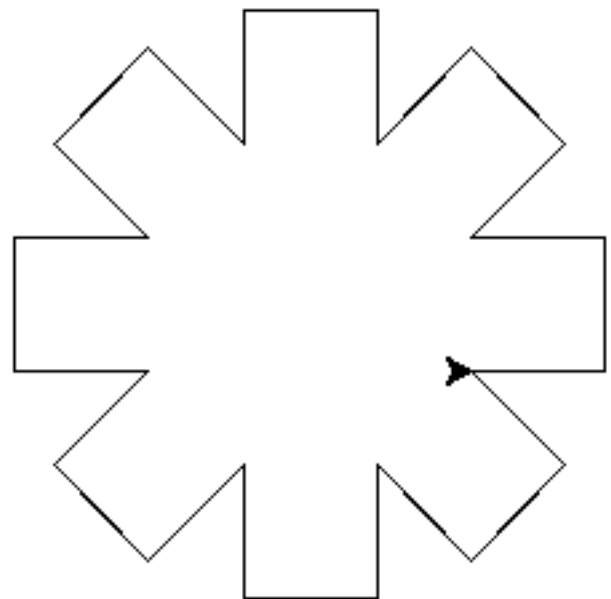
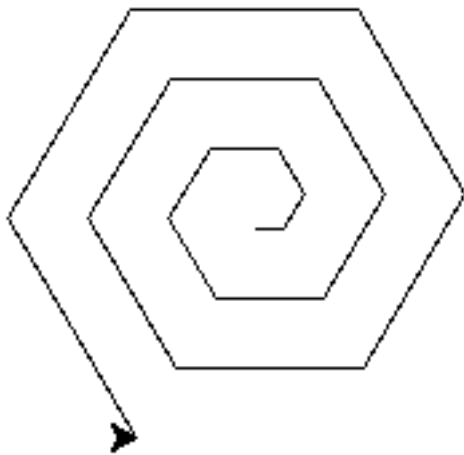
```
Nombre de notes ? -2  
  
Saisir un nombre > 0, nombre de notes ? 3  
  
Nouvelle note ? 10  
  
Nouvelle note ? 8  
  
Nouvelle note ? 5  
Moyenne : 7.67
```

**Sujet 3 : niveau moyen****Consignes**

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

**Exercice 1 *Boucles imbriquées et tortue***

Écrire avec le module `turtle` deux programmes permettant de réaliser les figures ci-dessous en un minimum de lignes (`from turtle import *`) compris.



**Sujet 4 : niveau moyen****Consignes**

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

 **Méthode *Concaténation de chaînes de caractères***

☞ On peut concaténer deux caractères avec l'opérateur + :

```
>>> '(' + ':'  
'(:'  
>>> '(' + '(' + '(' + '('  
'(((('
```

☞ Si on concatène plusieurs fois le même caractère on peut utiliser l'opérateur de duplication \*. On peut bien sûr combiner concaténation et duplication :

```
>>> '(' * 4  
'(((('   
>>> '(' * 4 + ')' * 4  
'(((())())'
```

 **Exercice 1 *Fonction et boucles***

Fournir des tests d'exécutions entre triple guillemets.

1. Écrire une fonction `carre(cote, caractere)` qui affiche un carré de côté passé en argument et rempli du caractère passé dans le second argument. La fonction doit vérifier que `cote` est un entier positif et provoquer une erreur sinon.

```
>>> carre(5, '@')
@@@@@
@@@@@
@@@@@
@@@@@
@@@@@
```

2. Écrire une fonction `triangle_rectangle_gauche(cote, caractere)` qui affiche un triangle rectangle avec l'angle droit à gauche, rempli du caractère passé en argument et avec `cote` caractères sur chaque côté adjacent à l'angle droit.

```
>>> triangle_rectangle_gauche(4, '@')
@
@@
@@@
@@@@
```

3. Écrire une fonction `triangle_rectangle_droite(cote, caractere)` qui affiche un triangle rectangle avec l'angle droit à droite, rempli du caractère passé en argument et avec `cote` caractères sur chaque côté adjacent à l'angle droit.


```
>>> triangle_rectangle_droit(5, '@')
  @
  @@
  @@@
  @@@@
  @@@@@
```

4. Écrire une fonction `triangle_isocele(base, caractere)` qui affiche un triangle isocèle rempli du caractère passé en argument et avec `base` caractères sur sa base.

```
>>> triangle_isocele(5, '@')
  @
  @@@
  @@@@@
  @@@@@@@
  @@@@@@@@@
```

**Sujet 5 : niveau difficile****Consignes**

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

 **Exercice 1 *Exercice 34 p. 46 du manuel Indice***



## Sujet 6 : niveau moyen

### Consignes


- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1\_NOM2\_Sujet10.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

### Exercice 1 *Pydefis*

Écrire avec le module `turtle` un programme résolvant le défi décrit sur cette page <https://pydefis.callicode.fr/defis/Queulorior/txt>.

**Sujet 7 : niveau difficile****Consignes**

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

 **Exercice 1 *Exercice 41 p. 47 du manuel Indice***

## Sujet 8 : niveau moyen

### Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

### Exercice 1 *Pydefis*

Écrire un programme résolvant le défi proposé sur cette page <https://pydefis.callicode.fr/defis/Escargot/txt>.

## Sujet 9 : niveau moyen

### Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1\_NOM2\_Sujet10.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

### Exercice 1 *Pydefis*

Écrire un programme résolvant le défi proposé sur cette page <https://pydefis.callicode.fr/defis/Herculito11Pommes/txt>.

### Exercice 2 *Pydefis*

Écrire un programme résolvant le défi proposé sur cette page <https://pydefis.callicode.fr/defis/SpymasterBomb/txt>

## Sujet 10 : niveau moyen

### Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1\_NOM2\_Sujet10.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

### Exercice 1 *Pydefis*

Écrire un programme résolvant le défi proposé sur cette page <https://pydefis.callicode.fr/defis/Herculito04Sanglier/txt>.

**Sujet 11 : niveau difficile****Consignes**

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format `NOM1_NOM2_Sujet10.py`
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 1 point pour l'activité de recherche en classe (différencié par binôme), 2 points si sujet difficile et 1 sinon, 3 points pour la qualité et la correction du code, 1 point pour la réalisation des objectifs fixés, 3 points pour la prestation orale (différencié par binôme).

 **Exercice 1 Pydefis**

Écrire un programme résolvant le défi proposé sur cette page [https://pydefis.callicode.fr/defis/C22\\_BwaCode01/txt](https://pydefis.callicode.fr/defis/C22_BwaCode01/txt).