

Colle 02

Cette deuxième colle vous fera écrire du code Python, et compléter du pseudo-code.

Fonctions arithmétiques

- Écrivez une fonction qui prend une liste Python `li` (une `list`) et renvoie la moyenne arithmétique de la liste `li` donnée en argument (i.e., $\sum_{j=0}^{n-1} li[j]/n$).
- Utilisez cette fonction pour calculer $M = \text{moyenne}(li)$, pour calculer la variance de la liste : c'est-à-dire la valeur $\text{Var}(li) = \frac{1}{n} \sum_{j=0}^{n-1} (li[j] - M)^2$.

Figures en mode texte

Écrivez une fonction qui affiche une figure comme celle-ci, pour $n \geq 6$ la hauteur donnée en argument de la fonction.

```
*      *
**     *
* *    *
* *    *
*      **
*      *
```

QCM

Entourez la bonne réponse. Il faudra pouvoir justifier.

Si on veut obtenir une liste de n couples de nombres flottants uniformément répartis entre $[-10, 10]$ en Python, quel(s) code(s) est/sont correct(s) ? Après avoir `import random`, on rappelle que `random.random()` donne un flottant aléatoirement réparti dans $[0, 1]$.

- `[random.random() * 20 - 10 for j in range(n)]`
- `[random.random(i) * 10 - 10 for i in range(n)]`
- `[random.random() * (-20) + 10] * n`
- `[-10 + random.random(20) for k in range(n)]`

Pseudo-code compléter

Corrigez le pseudo-code suivant en remplissant les trous `XXX`, pour obtenir une fonction qui teste si un entier `n` donné en argument est un nombre premier ou non.

```
Fonction estPremier(n : entier) -> booléen =
    maxDiviseur = partie entière inférieure de racineCarrée(n)
    Pour chaque diviseur = XXX à XXX (inclus) faire
        Si n mod diviseur != XXX Alors Renvoyer XXX
    Fin Pour
    Renvoyer XXX
```

Colle 02

Cette deuxième colle vous fera écrire du code Python, et compléter du pseudo-code.

Fonctions arithmétiques

- Écrivez une fonction qui prend une liste Python `li` (une `list`) et renvoie la moyenne arithmétique de la liste `li` donnée en argument (i.e., $\sum_{j=0}^{n-1} li[j]/n$).
- Utilisez cette fonction pour calculer $M = \text{moyenne}(li)$, pour calculer la variance de la liste : c'est-à-dire la valeur $\text{Var}(li) = \frac{1}{n} \sum_{j=0}^{n-1} (li[j] - M)^2$.

Figures en mode texte

Écrivez une fonction qui affiche une figure comme celle-ci, pour $n \geq 6$ la hauteur donnée en argument de la fonction.

```
*      *
*      *
*      *
*      *
*      *
*****
```

QCM

Entourez la bonne réponse. Il faudra pouvoir justifier.

Si on veut obtenir une liste de n couples de nombres flottants uniformément répartis entre $[-5, 5]$ en Python, quel(s) code(s) est/sont correct(s) ? Après avoir `import random`, on rappelle que `random.random()` donne un flottant aléatoirement réparti dans $[0, 1]$.

- `[random.random() * 10 - 5 for j in range(n)]`
- `[random.random(i) * 10 - 5 for i in range(n)]`
- `[random.random() * (-10) + 5] * n`
- `[-10 + random.random(10) for k in range(n)]`

Pseudo-code compléter

Corrigez le pseudo-code suivant en remplissant les trous `XXX`, pour obtenir une fonction qui teste si un entier `n` donné en argument est un nombre premier ou non.

```
Fonction estPremier(n : entier) -> booléen =
    maxDiviseur = partie entière inférieure de racineCarrée(n)
    Pour chaque diviseur = XXX à XXX (inclus) faire
        Si n mod diviseur != XXX Alors Renvoyer XXX
    Fin Pour
    Renvoyer XXX
```

Colle 02

Cette deuxième colle vous fera écrire du code Python, et compléter du pseudo-code.

Fonctions arithmétiques

- Écrivez une fonction qui prend une liste Python `li` (une `list`) et renvoie la moyenne arithmétique de la liste `li` donnée en argument (i.e., $\sum_{j=0}^{n-1} li[j]/n$).
- Utilisez cette fonction pour calculer $M = \text{moyenne}(li)$, pour calculer la variance de la liste : c'est-à-dire la valeur $\text{Var}(li) = \frac{1}{n} \sum_{j=0}^{n-1} (li[j] - M)^2$.

Figures en mode texte

Écrivez une fonction qui affiche une figure comme celle-ci, pour $n \geq 6$ la hauteur donnée en argument de la fonction.

```
*****
  *
 *
 *
*****
```

QCM

Entourez la bonne réponse. Il faudra pouvoir justifier.

Si on veut obtenir une liste de n couples de nombres flottants uniformément répartis entre $[-30, 30]$ en Python, quel(s) code(s) est/sont correct(s) ? Après avoir `import random`, on rappelle que `random.random()` donne un flottant aléatoirement réparti dans $[0, 1]$.

- `[random.random() * 30 - 60 for j in range(n)]`
- `[random.random(i) * 10 - 60 for i in range(n)]`
- `[random.random() * (-30) + 60] * n`
- `[-60 + random.random(30) for k in range(n)]`

Pseudo-code compléter

Corrigez le pseudo-code suivant en remplissant les trous `XXX`, pour obtenir une fonction qui teste si un entier `n` donné en argument est un nombre premier ou non.

```
Fonction estPremier(n : entier) -> booléen =
    maxDiviseur = partie entière inférieure de racineCarrée(n)
    Pour chaque diviseur = XXX à XXX (inclus) faire
        Si n mod diviseur != XXX Alors Renvoyer XXX
    Fin Pour
    Renvoyer XXX
```