Nom Prénom : ..... Date : .....
Première spécialité... **API :** Chapitre Informatique - Programmer en Python
Temps : 120 minutes ❑ Le sujet est à coller sur la copie double bien présentée

Produit de convolution

Consignes :

1. Faire un programme Python qui réalise un produit de convolution.
2. Afficher les probabilités des issues de la somme des faces d'un lancé de deux dés équilibrés à n faces.
3. Généraliser votre programme en augmentant le nombre de dés et en différenciant le nombre de faces par dé.

Toutes traces de recherche seront valorisées.
L'objectif de cette API est de construire un programme Python pour calculer les coefficients d'un développement d'un produit de deux polynômes. Cela s'appelle un produit de convolution.
Essayons de comprendre comment cela fonctionne.

Comprendre le concept ❑

❑ 1. Développer et réduire l'expression suivante :

$$(1 + 2x + 3x^2)(4 + 5x + 6x^2)$$

.....

Ordonner selon l'ordre croissant des puissances de x.

❑ 2. Compléter les tableaux suivants.

$\times$	1	$2x$	$3x^2$
$6x^2$			
$5x$			
4			

$\times$	1	2	3
6			
5			
4			

❑ 3. Sommer les nombres sur chacune des 5 diagonales. Que retrouve-t-on ?

.....

❑ 4. Déterminer une relation entre le nombre de lignes (et le nombre de colonnes) avec le nombre de diagonales.

.....

L'objectif est de la suite est de construire étape par étape le fameux programme python.

Programmer pas à pas ❑

❑ 5. On prend deux listes de nombres qui correspondent aux coefficients des deux polynômes. C'est à dire les listes `a = [1, 2, 3]` et `b = [4, 5, 6]`.

Donner la liste que l'on obtient à l'arrivée en classant les termes du polynôme dans le bon ordre ? .....

❑ 6. En utilisant une boucle `for`, parcourir la deuxième liste à l'envers.

❑ 7. À l'aide de deux boucles `for`, recréer le tableau précédent en affichant uniquement les coefficients des cases complétées.

❑ 8. L'objectif de la suite va être de stocker la somme des termes de chacune des diagonales dans une liste de `S` .

❑ 9. En s'aidant des question précédentes, créer la liste d'arrivée `S` composée de 0 en complétant la commande suivante :

```
S =  [0 for i range(.............)] #Longueur de S
```

❑ 10. Pour sommer les termes des diagonales, on va tester la condition `i + j == len(S)- 1` avec `i` et `j` l'indice des termes des listes `a` et `b`.

Quand on parcourt le tableau en Python, Si cette condition est vérifiée, ajouter au terme `S[i+j]` le terme parcouru dans le tableau.

❑ 11. Afficher la liste `S`.

❑ 12. Que remarque-t-on ? .....

Tester avec la condition `i + j == len(S)- 2`, `i + j == len(S)- 4` puis `i + j == len(S)- len(S)`

❑ 13. Modifier votre programme pour parcourir la liste `S` et y sommer les termes des diagonales.

On peut utiliser la condition `i + j == len(S)- (k + 1)` avec `k` l'indice de la liste `S`.

❑ 14. **Bonus :** Tester et modifier si besoin votre programme pour des listes `a` et `b` de longueurs différentes.

❑ 15. Afficher le résultat sous la forme d'un polynôme.

Utiliser une fonction utile pour vérifier ses résultats

```
from numpy import convolve
a = [1, 2, 3]
b = [4, 5, 6]
print(convolve(tuple(a), tuple(b)))
```

Tester votre programme pour le valider

	Entrée 1	**Entrée 2**	**Sortie**
Test 1	`a = [1, 1]`	`b = [2, 3]`	`S = [2, 5, 3]`
Test 2	`a = [1, 1, ,1, 1]`	`b =[1, 1, ,1, 1]`	`S = [1, 2, 3, 4, 3, 2, 1]`
Test 3	`a = [1, 2, 3]`	`b =[4, 5, 6, 45, 4]`	`S = [4, 13, 28, 72, 112, 143, 12]`
Test 4	`a = [1/2, 1/2]`	`b = [1/2, 1/2]`	`S = [0.25, 0.5, 0.25]`

Connaître les applications de ce programme

Le produit de convolution permet de :

- ➢ Lisser les courbes des fonctions.
- ➢ Flouter des photos.
- ➢ Trouver les probabilités issues de la somme d'un lancée de plusieurs dés à partir des listes des probabilités des issues de chaque dé.
- ➢ Vérifier les développements par la double distributivité.