

Programmation Vectorielle pour Calcul Matriciel

Accueil / Mes cours / PVCM / Sections / Exam 1/2 / PVCM_Test 3

Commencé le	Monday 12 May 2025, 09:00
État	Terminée
Terminé le	Monday 12 May 2025, 10:28
Temps mis	1 heure 27 min

Question **1**
Correct
Noté sur 2,00

🚩 Marquer la question

Trouver la plus grande valeur impaire d'une matrice. / Find the largest odd value of a matrix

Nombre de lignes maximum = 2

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def grand_impair(A):
    return np.max(A[A%2==1])
```

Question **2**

Correct

Noté sur 2,00

🚩 Marquer la question

On désire créer la matrice n x m qui a / We want to create a n by m matrix with

- [0,1,2,...n-1] en 1ere colonne, / in first column
- [1,2,3... n] en 2e colonne, / in 2nd column
- [2,3,4... n+1] en 3e colonne, / in 3rd column
- etc

`for` autorisé.

Nombre max de lignes = 2

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def matspe(n,m):
    return np.array([range(i, m+i) for i in range(n)])
```

Question **3**

Correct

Noté sur 2,00

🚩 Marquer la question

Ranger par ordre croissant les valeurs d'une matrice sans changer la structure de la matrice / Sort values of a matrix without changing its shape

$A[i,j] <= A[i+1,k]$ et $A[i,j] <= A[i,j+1]$ pour tout / for all ij,k

Nombre de lignes maximum = 2

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def range_mat(A):
    return np.sort(A.flatten()).reshape(A.shape)
```

Question **4**

Correct

Noté sur 3,00

🚩 Marquer la question

Ecrire une fonction qui échange les colonnes i et j de la matrice A / Swap columns i and j of matrix A

Nombre de lignes maximum = 3

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def xchange_col(A, i, j):
    A[:, [i,j]] = A[:, [j, i]]
    return A
```

Question **5**

Correct

Noté sur 3,00

🚩 Marquer la question

Remplacer toutes les valeurs négatives par leur opposé si cet opposé n'existe pas déjà dans la matrice. / Replace all negative values of a matrix by their opposite if it does not already exists in the matrix.

Nombre de lignes maximum = 5

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def presque_abs(A):
    mask = np.isin(A, -A[A>0])
    A = np.abs(A)
    A[mask] *= -1
    return A
```

Question **6**

Incorrect

Noté sur 8,00

🚩 Marquer la question

On désire calculer la position de la Lune après un certain temps.

On fait la simulation en 2D. Toutes les orbites sont des cercles dont le centre est l'objet précédant dans la liste. Tous les objets tournent dans le sens trigonométrique par rapport à leur centre.

- Le Soleil est en (0,0) et reste toujours à cette position.
- La Terre commence en (150550, 0) et fait un tour en 365 jours (exactement pour l'exercice). La distance avec le Soleil est de 150550.
- La Lune commence en (150934, 0) et fait un tour en 29 jours. La distance avec la Terre est de 384.

Ecrire la fonction mat_lune(t) qui retourne la matrice A qui permet d'obtenir la position de la Lune après un temps t exprimé en jours

```
pos = (mat_lune(t) @ np.array([150934, 0, 1]))[:2]
```

Attention, la Lune fait partie du système de la Terre, cela veut dire que si elle est dans l'alignement Soleil-Terre, elle le sera encore 29 jours plus tard.

/

We want to calculate the position of the Moon after a certain amount of time.

The simulation is done in 2D. All orbits are circles whose center is the previous object in the list. All objects rotate counterclockwise around their center.

- The Sun is at (0, 0) and always remains at this position.
- The Earth starts at (150550, 0) and completes one revolution in 365 days (exactly, for this exercise). The distance to the Sun is 150550.
- The Moon starts at (150934, 0) and completes one revolution in 29 days. The distance to the Earth is 384.

Write the function mat_lune(t) that returns the matrix A which gives the position of the Moon after a time t expressed in days:

```
pos = (mat_lune(t) @ np.array([150934, 0, 1]))[:2]
```

Note: The Moon is part of the Earth system, which means that if it is aligned with the Sun and the Earth, it will still be aligned 29 days later.

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

L'éditeur Ace n'est pas prêt. Recharger peut-être la page ?

Falling back to raw text area.

```
def mat_lune(t):
```