

Nom	
Prénom	
Groupe	

Note	
------	--

# Algorithmique Matrices

SUP S2 EPITA

## Examen B3

5 mars 2025

### Consignes (à lire) :

- Vous devez répondre directement **sur ce sujet**.
  - Répondez dans les espaces prévus, **les réponses en dehors ne seront pas corrigées**.
  - Aucune réponse au crayon de papier ne sera corrigée.
- La présentation est notée en moins, c'est à dire que vous êtes noté sur 20 et que les points de présentation (2 au maximum) sont retirés de cette note.
- Code :**
  - Tout code doit être écrit dans le langage Python (pas de C, CAML, ALGO ou autre).
  - **Tout code Python non indenté ne sera pas corrigé.**
  - Les seules classes, fonctions, méthodes que vous pouvez utiliser sont données en **annexe**.
  - Vos fonctions doivent impérativement respecter les exemples d'applications donnés.
  - Vous pouvez également écrire vos propres fonctions, dans ce cas elles doivent être documentées (on doit savoir ce qu'elles font).

Dans tous les cas, la dernière fonction écrite doit être celle qui répond à la question.

  - Comme d'habitude l'optimisation est notée. Si vous écrivez des fonctions non optimisées, vous serez notés sur moins de points.<sup>1</sup>

- Durée : 45min

## Annexes

### Fonctions et méthodes autorisées

Vous pouvez utiliser la méthode `append`, la fonction `len` sur les listes ainsi que la fonction `range` :

```
1 >>> L = []
2
3 >>> for i in range(5):
4     L.append(i)
5
6 >>> L
7 [0, 1, 2, 3, 4]
8
9 >>> len(L)
10 5
11
12 >>> for i in range(5, 10):
13     L.append(i)
14 >>> L
15 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

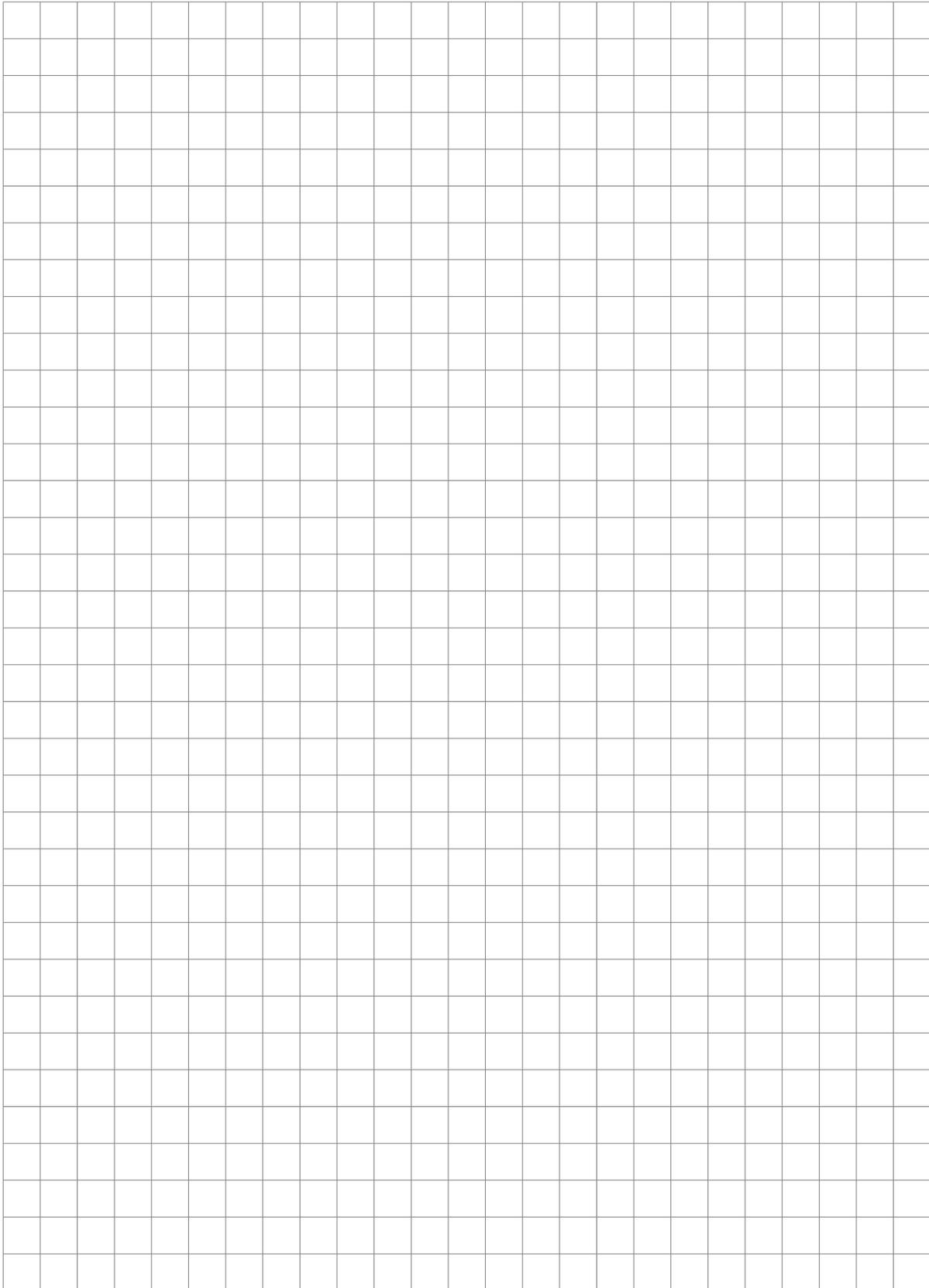
Aucun opérateur n'est autorisé sur les listes (+, \*, == ...).

---

1. Des fois, il vaut mieux moins de points que pas de points.

**Exercice 1 (Négatif – 4 points)**

Écrire la fonction `detect_neg(M)` avec `M` une matrice d'entiers non vide qui renvoie `True` si `M` contient au moins un nombre négatif, `False` sinon.

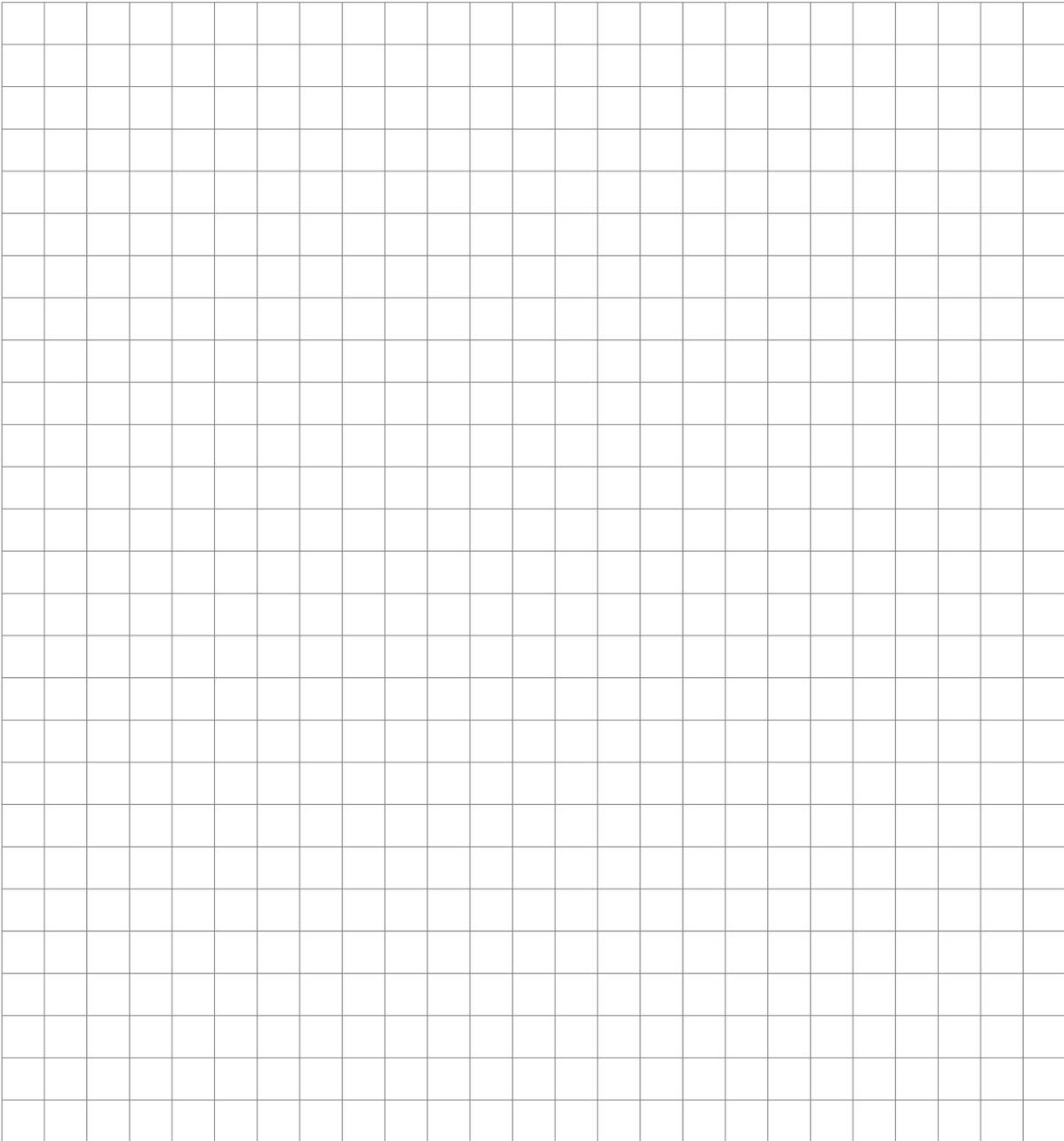


**Exercice 2 (Symétrique – 4 points)**

Écrire la fonction `build_symmetric(M)` qui construit et retourne une nouvelle matrice où les éléments de la matrice non vide `M` se trouvent dans la partie supérieure, et où la partie inférieure de la matrice est identique à la partie supérieure mais inversée verticalement.

*Exemple d'application :*

```
1 >>> build_symmetric([[19, 13, -6, -7, -1],
2                       [8, 20, 17, -2, 2],
3                       [-5, 15, -5, 7, -3]])
4 [[19, 13, -6, -7, -1],
5  [8, 20, 17, -2, 2],
6  [-5, 15, -5, 7, -3],
7  [-5, 15, -5, 7, -3],
8  [8, 20, 17, -2, 2],
9  [19, 13, -6, -7, -1]]
```



**Exercice 3 (Mystery – 2 points)**

```
1  def mystery(L):
2      res = []
3      max_col = 0
4      for i in range(len(L)):
5          line, col = L[i]
6          n = len(res)
7          for j in range(n, line+1):
8              res.append([])
9              nline = len(res[line])
10             for j in range(nline, col+1):
11                 res[line].append(-1)
12             res[line][col] = i
13             if col > max_col:
14                 max_col = col
15         for i in range(len(res)):
16             for j in range(len(res[i]), max_col+1):
17                 res[i].append(-1)
18         return res
```

Dessiner ci-dessous la matrice résultat de l'application de `mystery([(8, 2), (7, 2), (2, 5)])`.

