Concours National Commun (CNC), Maroc

Session 2025, filière TSI

Informatique

Durée : 2 heures

Ce que traite le sujet

- Exercice
- Partie I : apprentissage automatique
- Partie II : traitement d'images

Sujet reproduit sans retouche à partir de la page suivante.

Résumé des parties, chapitres de cours liés et autres années :
prepahelp.com/annales/cnc/tsi/2025/info

Prepa Help ne revendique aucun droit sur le sujet, propriété de l'organisateur du concours.

ⵜⴰⴳⵍⴷⵉⵜ ⵏ ⵍⵎⵖⵔⵉⴱ
ⵜⴰⵎⴰⵡⴰⵙⵜ ⵏ ⵜⵣⵣⵉⵡⵉⵏ ⴰⵏⵎⵓⵔⴰⵏ ⴰⵏⴰⵎⵓⵔ
ⵜⵣⵣⵉⵡⵉⵏ ⴰⵎⴰⵙⵙⴰⵏ ⴰⵏ ⵜⵙⵙⵏⵙⵡ

المملكة المغربية
وزارة التعليم العالي
والبحث العلمي والابتكار

Royaume du Maroc
Ministère de l'Enseignement Supérieur,
de la Recherche Scientifique et de l'Innovation

جامعة الحسن الثاني بالدار البيضاء
UNIVERSITE HASSAN II DE CASABLANCA

ENSEM

CONCOURS NATIONAL COMMUN

D'Admission dans les Établissements de Formation d'Ingénieurs et Établissements Assimilés

Session 2025

Épreuve d'Informatique

Filières : **TSI**

Durée : **2 heures**

Cette épreuve comporte **11 pages** au format A4, en plus de cette page de garde.

IMPORTANT
L'usage de la calculatrice et de tout système électronique ou informatique est interdit durant cette épreuve.

Les candidats sont informés que la précision des raisonnements algorithmiques ainsi que le soin apporté à la rédaction et à la présentation des copies seront des éléments pris en compte dans la notation. Il convient en particulier de rappeler avec précision les références des questions abordées. Si, au cours de l'épreuve, un candidat repère ce qui peut lui sembler être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Remarques générales :

- ✓ Cette épreuve est composée d'un exercice et de deux parties tous indépendants ;
- ✓ Toutes les instructions et les fonctions demandées seront écrites en Python ;
- ✓ Les questions non traitées peuvent être admises pour aborder les questions ultérieures ;
- ✓ Toute fonction peut être décomposée, si nécessaire, en plusieurs fonctions.

Important **: Le candidat doit impérativement commencer par traiter toutes les questions de l'exercice ci-dessous, et écrire les réponses dans les premières pages du cahier de réponses.**

Exercice : (4 points)

Q.1- Écrire la fonction ***factoriel (p)*** qui reçoit en paramètre un entier positif ***p***, et qui retourne la valeur de ***p!***.

NB : ***p! = 1*2*3* ... *p*** et ***0! = 1***

Q.2- Déterminer la complexité de la fonction ***factoriel (p)***, avec justification.

Q.3- Écrire la fonction ***som_fact (n)*** qui reçoit en paramètre un entier positif ***n***, et qui retourne la valeur de la somme suivante : ***0! + 1! + 2! + ... + n!***

Q.4- Écrire la fonction ***pairs (L)*** qui reçoit en paramètre une liste d'entiers positifs ***L***, et qui retourne le compte des nombres pairs dans ***L***.

Exemple : ***pairs ([15, 42, 7, 25, 42, 10])*** retourne ***3***

Q.5- Écrire la fonction ***som_impairs (L)*** qui reçoit en paramètre une liste d'entiers positifs ***L***, et qui retourne la valeur de la somme des nombres impairs dans ***L***.

Q.6- Écrire la fonction ***superieurs (L, x)*** qui reçoit en paramètres une liste de nombres ***L*** et un nombre ***x***. La fonction retourne une nouvelle liste qui contient les nombres de ***L*** qui sont supérieurs strictement à ***x***.

Q.7- Écrire la fonction ***list_sup (L, k)*** qui reçoit en paramètres une liste de nombres ***L*** et un nombre ***k***. La fonction retourne ***True*** si tous les nombres de ***L*** sont supérieurs strictement à ***k***, sinon, la fonction retourne ***False***.

Partie I :

Apprentissage automatique

Prix des téléphones mobiles

Une nouvelle entreprise de téléphonie mobile veut livrer un combat difficile aux grandes entreprises comme Apple, Samsung, etc.

Dans le but d'estimer le prix des mobiles qu'elle crée, cette nouvelle entreprise a collecté des données de diverses entreprises de téléphonie mobile. Cet ensemble de données permet d'identifier les facteurs sous-jacents qui affectent les prix des téléphones mobiles. En analysant les relations entre ces fonctionnalités et leurs gammes de prix respectives, on peut obtenir des informations précieuses sur la dynamique du marché des téléphones mobiles

Les données collectées pour chaque téléphone mobile sont composées de **15** variables (caractéristiques), dont les **14** premières sont des variables prédictives. La dernière est une variable cible :

1. ***puiss_batterie*** : Puissance de la batterie
2. ***vitesse_micropr*** : Vitesse du microprocesseur
3. ***cam_fr*** : Capacité de la caméra frontale
4. ***mem_int*** : Capacité de la mémoire interne en giga-octet
5. ***profondeur*** : Profondeur du téléphone mobile
6. ***poids*** : Poids du téléphone mobile
7. ***cœurs*** : Nombre de cœurs d'un processeur
8. ***cam_pr*** : Capacité de l'appareil photo principal
9. ***resolution_htr*** : Hauteur de résolution des pixels
10. ***resolution_lrg*** : Largeur de résolution des pixels
11. ***mémoire*** : Capacité de la mémoire vive en giga-octet
12. ***ecran_htr*** : Hauteur de l'écran du téléphone mobile
13. ***ecran_lrg*** : Largeur de l'écran du téléphone mobile
14. ***max_charge*** : La durée la plus longue d'une seule charge de batterie
15. ***prix*** : Le prix du mobile (variable cible)

Base de données et langage SQL :

Les données collectées par la nouvelle entreprise sont rangées dans une base de données composée de deux tables **'Entreprise'** et **'mobile'** dont les structures sont les suivantes :

- **Entreprise (code** (texte) **, nom** (texte), **pays** (texte) **)** ;
- L'attribut **code** est la clé primaire de la table **'Entreprises'** ;
- **Mobiles (codEntrp** (texte), **puiss_batterie** (entier), **vitesse_micropr** (réel), **cam_fr** (entier), **mem_int** (entier), **profondeur** (réel), **poids** (entier), ***cœurs*** (entier), **cam_pr** (entier), **resolution_htr** (entier), **resolution_lrg** (entier), **mémoire** (entier), **ecran_htr** (entier), **ecran_lrg** (entier), **max_charge** (entier), **prix** (réel) **)**
- Dans la table **'Mobile'**, l'attribut **codEntrp** est une clé étrangère qui fait référence à la clé primaire de la table **'Entreprise'**.

Exemples :

Entreprise		
code	*nom*	*pays*
apl	Apple	USA
sms	Samsung	Corée sud
hw	Huawei	Chine
xm	Xiaomi	Chine
...	...	...

Mobiles															
codEntrp	puiss_batterie	vitesse_micropr	cam_fr	mem_int	profondeur	poids	coeures	cam_pr	resolution_htr	resolution_lrg	mémoire	ecran_htr	ecran_lrg	max_charge	prix
sms	842	2.2	1	7	0.6	188	2	2	20	756	2549	9	7	19	1400
hw	510	2	5	45	0.9	168	6	16	483	754	3919	19	4	2	3500
apl	563	0.5	2	41	0.9	145	5	6	1263	1716	2603	11	2	9	2300
xm	615	2.5	0	10	0.8	131	6	9	1216	1768	2769	16	8	11	2000
xm	1821	1.2	13	44	0.6	141	2	14	1208	1212	1411	8	2	15	1200
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
apl	1021	0.5	0	53	0.7	136	3	6	905	1988	2631	17	3	7	2500

Q.8 – Écrire en algèbre relationnelle, la requête qui donne pour résultat les codes et les noms des entreprises du pays *'Chine'*.

Q.9 – Écrire en langage SQL, la requête qui donne pour résultat la capacité de la caméra frontale et la capacité de l'appareil photo principal des mobiles de l'entreprise de nom *'Honor'*, triés dans l'ordre décroissant de la capacité de l'appareil photo principal.

Q.10 – Écrire en langage SQL, la requête qui donne pour résultat la plus grande et la plus petite capacité de la mémoire vive des mobiles.

Q.11 – Écrire en langage SQL, la requête qui donne pour résultat les codes des entreprises, le compte des mobiles pour chaque entreprise. Le résultat doit être trié dans l'ordre décroissant des comptes des mobiles.

Q.12 – Écrire en langage SQL, la requête qui donne pour résultat les noms des entreprises sans répétition, qui fabriquent des mobiles dont la capacité de la mémoire vive est inférieure à la moyenne de la capacité la mémoire vive de tous les mobiles.

Q.13 – Écrire en langage SQL, la requête qui donne pour résultat les noms des entreprises telles que la moyenne des prix des mobiles de chaque entreprise est supérieure à *7500*. Le résultat doit être trié dans l'ordre décroissant de la moyenne des prix.

Dans la suite de cette partie, les données collectées sont rangées dans une liste ***M***, déclarée variable globale. Chaque élément ***Mi*** est une liste qui représente les **15** attributs d'un téléphone mobile.

```
M = [ [ 842, 2.2, 1, 7, 0.6, 188, 2, 2, 20, 756, 2549, 9, 7, 19, 1400 ],
      [ 510, 2, 5, 45, 0.9, 168, 6, 16, 483, 754, 3919, 19, 4, 2, 3500 ],
      [ 563, 0.5, 2, 41, 0.9, 145, 5, 6, 1263, 1716, 2603, 11, 2, 9, 2300 ],
      ...
      [ 1021, 0.5, 0, 53, 0.7, 136, 3, 6, 905, 1988, 2631, 17, 3, 7, 2500 ] ]
```

Dans chaque liste ***Mi*** :

- L'indice ***0*** correspond à l'attribut ***puiss_batterie*** des téléphones mobiles. Cet attribut est représenté par les nombres : **842, 510, 563, 615, 1821, ...1021**
- L'indice ***1*** correspond à l'attribut ***vitesse_micropr*** des téléphones mobiles. Cet attribut est représenté par les nombres : **2.2, 2, 0.5, 2.5, 1.2, ..., 0.5**
- ...
- L'indice ***14*** correspond à l'attribut ***prix*** des téléphones mobiles. Cet attribut est représenté par les nombres : **1400, 3500, 2300, 2000, 1200, ..., 2500**

Matrice de corrélation :

Pour évaluer la dépendance entre les attributs en même temps, on utilise la ***matrice de corrélation***.

La matrice de corrélation est une matrice symétrique qui contient les coefficients de corrélation entre chaque attribut et les autres attributs.

L'interprétation du coefficient de corrélation ***r*** dépend de sa valeur :

- ***-1<r<0 :*** la corrélation entre les deux attributs est négative, donc lorsqu'un attribut augmente l'autre diminue. Plus la valeur est proche de **-1**, plus les attributs sont liés négativement.
- ***0<r<1 :*** la corrélation entre les deux attributs est positive, plus la valeur est proche de **+1**, plus la relation entre les attributs est forte. Dans ce cas, un attribut a tendance à augmenter en valeur lorsque l'autre augmente également.
- ***r = 0 :*** la corrélation entre les deux attributs est très faible, en effet, la relation linéaire entre eux est nulle. Mais les deux attributs pourraient avoir une relation non linéaire.
- ***r = 1 :*** les deux attributs ont une corrélation positive parfaite, c'est-à-dire qu'ils ont une relation linéaire positive.
- ***r = -1 :*** les deux attributs ont une corrélation négative parfaite, c'est-à-dire qu'ils ont une relation linéaire négative.

Il existe différentes méthodes pour calculer les coefficients de corrélation : méthode de Pearson, méthode de Kendall et celle de Spearman.

Méthode de Pearson :

Le coefficient de corrélation de deux attributs est égal au quotient entre la covariance des deux attributs et la racine carrée du produit de la variance de chaque attribut.

Q.14- Écrire la fonction ***moyenne (j)*** qui reçoit en paramètre un entier ***j*** qui représente l'indice d'un attribut dans la liste des téléphones mobiles ***M***. La fonction retourne la moyenne de cet attribut.

$$moyenne\ (j) = \frac{1}{n} * \sum_{i=0}^{n-1} M_{ij} \qquad n\ est\ la\ taille\ de\ M$$

Exemple :

moyenne (5) retourne le nombre **140.249**

Q.15- Écrire la fonction ***variance (j)*** qui reçoit en paramètre un entier ***j*** qui représente l'indice d'un attribut dans la liste des téléphones mobiles ***M***. La fonction retourne la variance de cet attribut.

$$variance\ (j) = \frac{1}{n} * \sum_{i=0}^{n-1} (M_{i,j} - m)^2$$

n est la taille de ***M***, et ***m*** est la moyenne de l'attribut d'indice ***j*** dans ***M***

Exemple :

variance (5) retourne le nombre **1252.5089989999988**

Q.16- Écrire la fonction ***covariance (i, j)*** qui reçoit en paramètres deux entiers ***i*** et ***j*** qui représentent les indices de deux attributs dans ***M***. La fonction retourne la covariance de ces deux attributs.

$$covariance\ (i,j) = \frac{1}{n} * \sum_{k=0}^{n-1} (M_{k,i} - mi) * (M_{k,j} - mj)$$

n est la taille de ***M***, et ***mi*** et ***mj*** sont respectivement est les moyennes des attributs d'indices ***i*** et ***j*** dans ***M***.

Exemple :

covariance (2, 5) retourne le nombre **3.627934499999999**

Q.17- Écrire la fonction ***coeff_correlation (i, j)*** qui reçoit en paramètres deux entiers ***i*** et ***j*** qui représentent les indices de deux attributs dans ***M***. En utilisant la méthode de *Pearson*, la fonction retourne le coefficient de corrélation correspondant à ces deux attributs.

$$coef_correlation(i,j) = \frac{covariance\ (i,j)}{\sqrt{variance(i) * variance(j)}}$$

Exemple :

coeff_correlation (2, 5) retourne le nombre **0.02361801709681716**

Q.18- Écrire la fonction ***matrice_correlation ( )*** qui retourne la matrice de corrélation correspondante à la liste des téléphones mobiles ***M***.

Exemple :

matrice_correlation ( ) retourne la matrice de corrélation suivante :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1.0	0.011	0.033	-0.004	0.034	0.002	-0.03	0.031	0.015	-0.008	-0.001	-0.03	-0.021	0.053	0.201
1	0.011	1.0	-0.0	0.007	-0.014	0.012	-0.006	-0.005	-0.015	-0.009	0.003	-0.029	-0.007	-0.011	-0.007
2	0.033	-0.0	1.0	-0.029	-0.002	0.024	-0.013	0.645	-0.01	-0.005	0.015	-0.011	-0.012	-0.007	0.022
3	-0.004	0.007	-0.029	1.0	0.007	-0.034	-0.028	-0.033	0.01	-0.008	0.033	0.038	0.012	-0.003	0.044
4	0.034	-0.014	-0.002	0.007	1.0	0.022	-0.004	0.026	0.025	0.024	-0.009	-0.025	-0.018	0.017	0.001
5	0.002	0.012	0.024	-0.034	0.022	1.0	-0.019	0.019	0.001	0.0	-0.003	-0.034	-0.021	0.006	-0.03
6	-0.03	-0.006	-0.013	-0.028	-0.004	-0.019	1.0	-0.001	-0.007	0.024	0.005	-0.0	0.026	0.013	0.004
7	0.031	-0.005	0.645	-0.033	0.026	0.019	-0.001	1.0	-0.018	0.004	0.029	0.005	-0.024	0.015	0.034
8	0.015	-0.015	-0.01	0.01	0.025	0.001	-0.007	-0.018	1.0	0.511	-0.02	0.06	0.043	-0.011	0.149
9	-0.008	-0.009	-0.005	-0.008	0.024	0.0	0.024	0.004	0.511	1.0	0.004	0.022	0.035	0.007	0.166
10	-0.001	0.003	0.015	0.033	-0.009	-0.003	0.005	0.029	-0.02	0.004	1.0	0.016	0.036	0.011	0.917
11	-0.03	-0.029	-0.011	0.038	-0.025	-0.034	-0.0	0.005	0.06	0.022	0.016	1.0	0.506	-0.017	0.023
12	-0.021	-0.007	-0.012	0.012	-0.018	-0.021	0.026	-0.024	0.043	0.035	0.036	0.506	1.0	-0.023	0.039
13	0.053	-0.011	-0.007	-0.003	0.017	0.006	0.013	0.015	-0.011	0.007	0.011	-0.017	-0.023	1.0	0.022
14	0.201	-0.007	0.022	0.044	0.001	-0.03	0.004	0.034	0.149	0.166	0.917	0.023	0.039	0.022	1.0

Dans la matrice de corrélation, on remarque que :

- L'attribut *mémoire* (l'attribut d'indice ***10***) a une forte corrélation positive avec la variable cible *prix*
- L'attribut *poids* (l'attribut d'indice ***5***) a une forte corrélation négative avec la variable cible *prix*

Q.19- Écrire la fonction ***attribut (C)*** qui reçoit en paramètre une matrice de corrélation ***C*** et qui retourne l'indice de l'attribut qui a une forte corrélation positive avec la variable cible.

Exemple : ***attribut (C)*** retourne l'indice ***10***

Algorithme KNN :

L'algorithme KNN est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage supervisé, et qui peut être utilisé pour résoudre les problèmes de régression.

L'algorithme KNN reçoit un ensemble de données qui est étiqueté avec des valeurs de sorties correspondantes sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction. Cet algorithme pourra par la suite être utilisé sur de nouvelles données afin de prédire leurs valeurs de sorties correspondantes.

Pour la classification d'un nouveau téléphone mobile ***X***, le principe de l'algorithme ***KNN*** est le suivant :

- Pour mesurer la similarité, on définit une fonction qui calcule la distance entre le nouveau téléphone mobile ***X*** et un autre téléphone mobile de ***M*** ;
- On calcule toutes les distances entre chaque téléphones mobiles de ***M***, le téléphone mobile ***X*** ;
- On choisit un nombre entier strictement positif ***k*** ;
- On retourne la moyenne des ***prix*** des ***k*** téléphones mobiles de ***M*** les proches au nouveau téléphone mobile ***X***.

Q.20- Pour la mesure de similarité, on propose d'utiliser la distance de *Manhattan*.

La liste ***X*** représente un nouveau téléphone mobile, et la liste ***e*** représente un téléphone mobile de ***M***. La distance de *Manhattan* entre ***X*** et ***e*** est définie par la formule suivante :

$$\boldsymbol{distance}\ (X, e) = \sum_{i=0}^{13} |X_i - e_i|$$

Écrire la fonction ***distance (X, e)*** qui reçoit en paramètres les deux téléphones mobiles ***X*** et ***e*** et qui retourne la distance euclidienne entre ***X*** et ***e***.

Exemple :

X = [594, 0.8, 2, 28, 0.1, 132, 5, 3, 1011, 1263, 3087, 16, 4, 4]

e = [842, 2.2, 1, 7, 0.6, 188, 2, 2, 20, 756, 2549, 9, 7, 19, 1400]

distance (X, e) retourne **2392.9**

Q.21- Écrire la fonction ***indices ( )*** qui retourne la liste ***D*** des indices des éléments de ***M***.

Exemple :

indices ( ) retourne la liste ***D*** = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...]

Q.22- Écrire la fonction ***tri (D, X)*** qui reçoit en paramètres la liste ***D*** des indices des éléments de ***M*** et la liste ***X*** qui représente un nouveau téléphone mobile. La fonction trie les éléments D_i de ***D*** dans l'ordre croissant de la distance *Manhattan* entre ***X*** et chaque élément de ***M*** d'indice D_i. (Ne pas utiliser la fonction *sorted()* et la méthode *sort()*)

Exemple :

X = [594, 0.8, 2, 28, 0.1, 132, 5, 3, 1011, 1263, 3087, 16, 4, 4]

Après l'appel de la fonction ***tri (D, X)***, on obtient la liste suivante :

[55, 14, 49, 72, 17, 9, 178, 21, 141, 42, 11, 50, 177, 113, 19, 156, 39, 10, 16, 88, 332, 38, ...]

- Le téléphone mobile M_{55} est le premier plus proche téléphone mobile à ***X*** ;
- Le téléphone mobile M_{14} est le deuxième plus proche téléphone mobile à ***X*** ;
- Le téléphone mobile M_{49} est le troisième plus proche téléphone mobile à ***X*** ;
- ...

Q.23- Écrire la fonction ***knn (k, X)*** qui reçoit en paramètres un entier strictement positif ***k***, un nouveau téléphone mobile ***X*** et la liste des téléphones mobiles ***M***. En utilisant l'algorithme ***KNN***, la fonction retourne la valeur du ***prix*** prédite pour le nouveau téléphone mobile ***X***.

Exemple :

X = [594, 0.8, 2, 28, 0.1, 132, 5, 3, 1011, 1263, 3087, 16, 4, 4]

knn (15, X) retourne ***2385.4***

Q.24- On suppose que la liste ***T*** représente les **14** attributs des téléphones mobiles crées par la nouvelle entreprise : Chaque élément T_i est une liste qui représente les **14** attributs d'un nouveau téléphone mobile.

Exemple :

T = [[742, 0.5, 0, 36, 0.7, 139, 3, 1, 1161, 1288, 292, 8, 0, 6],
[1306, 2.1, 2, 33, 0.4, 174, 3, 9, 867, 1258, 2521, 6, 5, 16],
[1424, 2.9, 0, 20, 0.6, 180, 5, 0, 181, 1199, 2658, 15, 0, 19],
...]

Écrire la fonction ***estimation_prix (k, T)*** qui reçoit en paramètres un entier strictement positif ***k***, et la liste des téléphones mobiles ***T***. En utilisant l'algorithme ***KNN***, la fonction retourne une nouvelle liste ***P*** telle que : chaque élément P_i est le prix prédit pour le téléphone T_i.

Partie II :

Agrandissement d'image

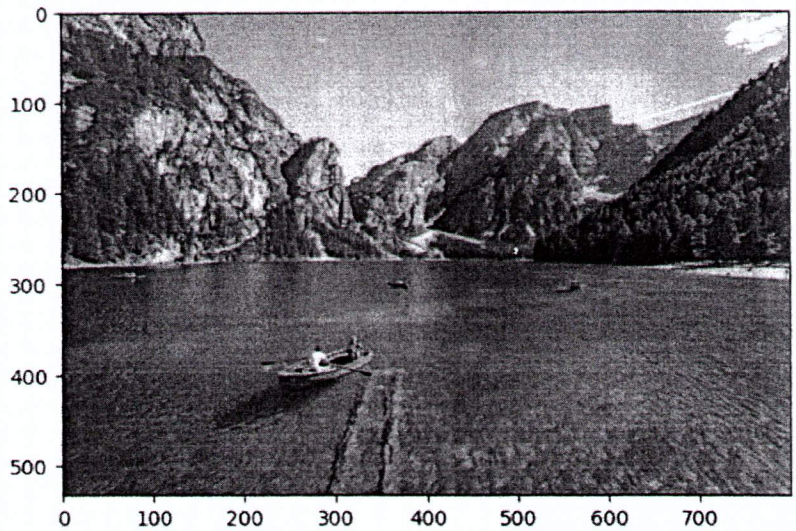
Interpolation bilinéaire

Dans cette partie, on suppose que le module ***numpy*** est importé :

```
import numpy as np
```

Une image numérique en **niveaux de gris** est représentée par une matrice à deux dimensions. Chaque élément de la matrice est appelé **pixel**. La valeur de chaque pixel est un entier compris entre **0** et **255**, ainsi l'image est codée sur **256** niveaux de gris. Le nombre **0** correspond au noir, et le nombre **255** correspond au blanc. Et chacun des autres nombres **1**, **2**, **3**,..., **253**, **254** correspond à un niveau de gris entre le noir et le blanc.

Exemple :



L'image en niveaux de gris ci-dessus, est représentée par la matrice ***M*** suivante *(533 lignes, 800 colonnes)* :

```
[[129 139  77 100 127  78  86 105  98 103 121 106 100 131 142  61  ... ]
 [ 78  78  98 106 127 105  44  60  93  84 103  93 126 158 174 106  ... ]
 [ 60  66 125 129 142 136  57  51  58  74  70  96  91 150 152 145  ... ]
 [ 84 110 129 141 141 132 112  77 111 122 127  81  66  78 129 173  ... ]
 [ 64 100 100 124 130 136 165 129  60 117 134 145  83 100  76 121  ... ]
  ...
```

Pour plus de clarté, dans la suite de cette partie, on utilisera cette image dans les exemples des fonctions.

Le redimensionnement est une transformation applicable à une image numérique qui consiste à en modifier la taille, que ce soit pour l'agrandir ou pour la rétrécir, comme le ferait un zoom.

L'agrandissement d'une image se fait par interpolation :

- Dans l'image, on ajoute de nouvelles colonnes (ou lignes) de pixels ;
- Ensuite, en utilisant le principe de l'interpolation, on calcul la valeur de chaque nouveau pixel en fonction de ses pixels voisins.

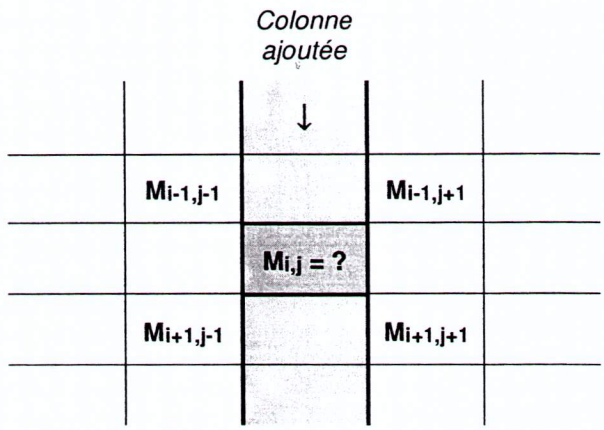
*L'**interpolation bilinéaire** est l'une des méthodes d'interpolation pour l'agrandissement d'une image. Elle prend en compte les quatre voisins du nouveau pixel que l'on calcule, il applique une fonction bilinéaire (d'ordre 2) sur les quatre voisins, puis applique le résultat sur le pixel interpolé.*

Principe de l'interpolation bilinéaire pour ajouter une colonne :

On commence par ajouter une colonne à un indice ***k*** compris strictement entre l'indice de la première colonne et l'indice de la dernière colonne de l'image.

Dans la nouvelle colonne ajoutée, on considère un pixel $M_{i,j}$ tel que :

- L'indice ***i*** est compris strictement entre l'indice de la première ligne de l'image et l'indice de la dernière ligne de l'image
- L'indice ***j*** est compris strictement entre l'indice de la première colonne de l'image et l'indice de la dernière colonne de l'image



La valeur du nouveau pixel $M_{i,j}$ est la valeur interpolée.

L'interpolation bilinéaire peut s'interpréter comme une succession de deux interpolations linéaires, une dans chaque direction. Elle peut se mettre sous la forme :

$$M_{i,j} = \frac{1}{2} * dfi * + \frac{1}{2} * dfj * + \frac{1}{4} * dfij + M_{i+1,j-1}$$

Avec :

$$dfi = M_{i-1,j-1} - M_{i+1,j-1}$$
$$dfj = M_{i+1,j+1} - M_{i+1,j-1}$$
$$dfij = M_{i+1,j-1} + M_{i-1,j+1} - M_{i-1,j-1} - M_{i+1,j+1}$$

Cas particuliers :

- ✓ Si le pixel $M_{i,j}$ à interpoler se trouve sur la première ligne de l'image, alors ses 4 voisins sont les suivants :

	$M_{i,j-1}$	$M_{i,j} = ?$	$M_{i,j+1}$	
	$M_{i+1,j-1}$		$M_{i+1,j+1}$	

- ✓ Si le pixel $M_{i,j}$ à interpoler se trouve sur la dernière ligne de l'image, alors ses 4 voisins sont les suivants :

	$M_{i-1,j-1}$		$M_{i-1,j+1}$	
	$M_{i,j-1}$	$M_{i,j} = ?$	$M_{i,j+1}$	

Q.25- Écrire la fonction ***interpolation_bilineaire (M, i, j)*** qui reçoit en paramètres une matrice ***M*** qui représente une image en niveaux de gris, et ***i*** et ***j*** sont les indices d'un pixel dont on veut interpoler la valeur. La fonction retourne la valeur interpolée du pixel $M_{i,j}$.

Q.26- Écrire la fonction ***ajout_colonne (M, k)*** qui reçoit en paramètres une matrice ***M*** qui représente une image en niveaux de gris, et entier **k** compris strictement entre l'indice de la première colonne et l'indice de la dernière colonne de ***M***. En utilisant l'interpolation bilinéaire, la fonction ajoute une colonne dans ***M*** à l'indice ***k***.

Exemple :

ajout_colonne (M, 2) ajoute une colonne à l'indice ***2*** et retourne la matrice suivante :

```
[[129  139  119   77  100  127  ...]
 [ 78   78  108   98  106  127  ...]
 [ 60   66  108  125  129  142  ...]
 [ 84  110  104  129  141  141  ...]
 [ 64  100  117  100  124  130  ...]
 ...  ...  ...  ...  ...  ...
 [ 96   92   85   86   79   76  ...]]
```

Q.27- Écrire la fonction ***ajout_ligne (M, k)*** qui reçoit en paramètres une matrice ***M*** qui représente une image en niveaux de gris, et entier **k** compris strictement entre l'indice de la première ligne et l'indice de la dernière ligne de ***M***. En utilisant l'interpolation bilinéaire, la fonction ajoute dans ***M*** une ligne à l'indice ***k***.

Exemple :

ajout_ligne (M, 3) ajoute une ligne à l'indice ***3*** et retourne la matrice suivante :

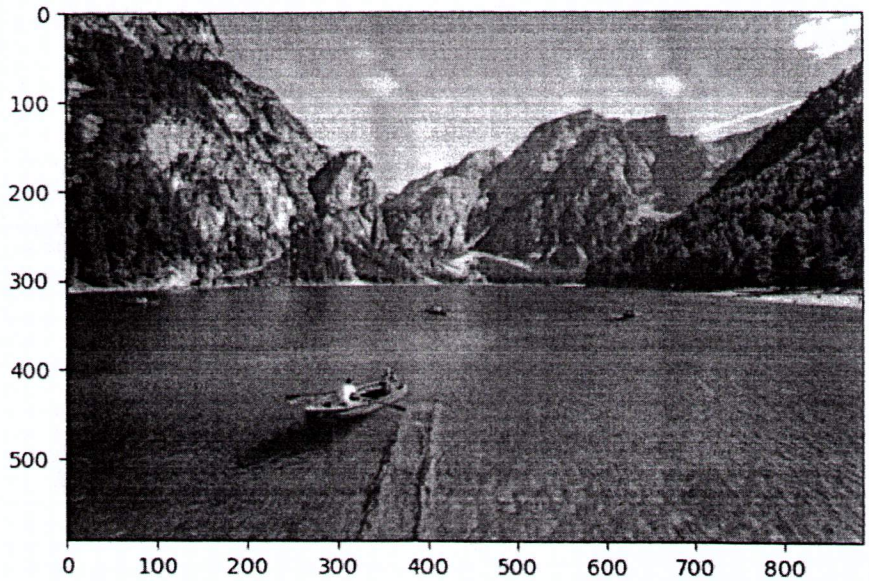
```
[[ 129  139   77  100  127  ...]
 [ 78   78    98  106  127  ...]
 [ 60   66   125  129  142  ...]
 [ 62   87   104  124  131  ...]
 [ 84  110   129  141  141  ...]
 [ 64  100   100  124  130  ...]
 ...   ...   ...  ...  ...
 [ 96   92    86   79   76  ...]]
```

Q.28- Écrire la fonction ***zoom (M, p)*** qui reçoit en paramètres la matrice ***M*** qui représente une image en niveaux de gris, et ***p*** un entier strictement positif. En utilisant l'interpolation bilinéaire, la fonction effectue l'agrandissement de l'image ***M*** suivant :

- ✓ Ajouter dans ***M*** les colonnes aux indices : ***p, 2*p+1, 3*p+2, 4*p+3, 5*p+4,*** ...
- ✓ Ajouter dans ***M*** les lignes aux indices : ***p, 2*p+1, 3*p+2, 4*p+3, 5*p+4,*** ...

Exemple :

Après l'appel de la fonction ***zoom (M, 10)***, on obtient l'image suivante :



############ **FIN** ############

Travailler les chapitres de ce sujet

Prepa Help suit le programme de TSI chapitre par chapitre : cours détaillé, méthodes, pièges, exercices et problèmes corrigés pas à pas, tirés des concours. L'essai gratuit donne accès aux chapitres offerts de ta filière, sans carte bancaire.

Chapitres liés à ce sujet

- Bases de Python : expressions, types, instructions, fonctions
- Complexité des algorithmes itératifs et récursifs
- Tri par fusion et tri rapide

Ouvrir l'essai gratuit

Scanne le code ou clique sur le bouton.

prepahelp.com/#essai

Une question sur le concours ou sur ta filière : **écrire sur WhatsApp**

Les autres années et épreuves : prepahelp.com/annales/cnc/tsi