

Section : Sciences de l'Informatique

Durée : 1h30

Epreuve : Algorithmique et programmation

Date : Réserve

Important :

1. Dans le dossier **Bac2014** de la racine **C:** de votre poste, créez un dossier de travail ayant pour nom votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer au fur et à mesure tous les fichiers solutions au problème posé.
2. Vérifiez à la fin de l'épreuve que tous les fichiers que vous avez créés sont dans votre dossier de travail.
3. Une solution modulaire au problème posé est exigée.

On se propose de remplir une matrice carrée **M** d'ordre **N** par des **nombre premiers** de l'intervalle $[2, 99]$ et de rechercher les lignes et les colonnes de cette matrice dont les éléments sont ordonnés dans l'ordre croissant ou décroissant.

Exemple :

Pour la matrice carrée **M** d'ordre **N=5** ci-contre, les lignes et les colonnes dont les éléments sont ordonnés dans l'ordre croissant ou décroissant sont :

	1	2	3	4	5
1	2	73	79	5	23
2	5	29	37	61	67
3	23	17	43	53	59
4	29	23	47	7	89
5	31	29	23	19	17

- La ligne 2 formée par les éléments 5 - 29 - 37 - 61 - 67
- La ligne 5 formée par les éléments 31 - 29 - 23 - 19 - 17
- La colonne 1 formée par les éléments 2 - 5 - 23 - 29 - 31

Travail demandé :

Ecrire un programme Pascal permettant de :

- remplir aléatoirement une matrice **M** d'ordre **N** donné (avec $4 < N < 20$) par des nombres premiers de l'intervalle $[2, 99]$.
- stocker dans chacune des lignes du fichier texte "**Result.txt**", la lettre "**L**" suivie par le numéro de la ligne de la matrice ou la lettre "**C**" suivie par le numéro de la colonne de la matrice **M** vérifiant la propriété citée ci-dessus, suivi d'un astérisque "*", suivi des éléments de cette ligne ou de cette colonne séparés par des tirets "-", comme le montre le formatage suivant :

Exemple : pour la matrice **M** de l'exemple ci-dessus, le contenu du fichier "**Result.txt**" sera :

- L2*5-29-37-61- 67
- L5*31-29-23-19-17
- C1*2-5-23-29-31

Grille d'évaluation

Questions	Nombre de points
• Décomposition en modules	2
• Appels des modules	2
• Si exécution et tests réussis avec respect des contraintes	16
Sinon	
✓ Structures de données adéquates au problème posé	2,5
✓ Saisie de N et remplissage de la matrice M	5
✓ Remplissage du fichier " Result.txt "	8,5