**Corrigé DM SIO1**

Exercice 1 : Distances InterVilles

Algorithme «  Distances Intervilles »

Variable indicées : Distance [10,10] de type numérique

Ville [10] de type CC

Variables : CC = VilleDép, VilleArr, Rep

Numérique = I,J, Dist

Début

(\* On considère que la matrice et le vecteur sont pré-remplis\*)

Répéter

Afficher »Quelle est la ville de départ ? »

Saisir VilleDép

Afficher »Quelle est la ville d’arrivée ? »

Saisir VilleArr

I🡨1

Tant que (I <= 10) et (Ville{I] <> VilleDép)

I 🡨 I+1

FinTantque

Si I > 10 Alors Afficher »La ville de départ saisie est inexistante «

Sinon J🡨1

Tant que (J <= 10) et (Ville{I] <> VilleArr)

J 🡨 J+1

FinTantque

Si J>10 Alors Afficher »Ville d’arrivée saisie inexistante «

Sinon Si J > I Alors Dist 🡨 Distance[ I, J ]

Sinon Dist 🡨 Distance[ J, I ]

Fsi

Afficher « La distance entre les 2 villes est de « , Dist, »KMS

Fsi

Fsi

Afficher «  Autre distance ? (Oui/Non) »

Saisir Rep

Jusqu’à Rep = « Non »

Fin

Exercice 2 : Méthode de tri par comptage

1. Jeu d’essai

Soit le vecteur Tab de 6 cases à trier :

Tab a une tille de 6 cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 7 | 1 | 2 | 4 | 3 |

Les différentes étapes sont :

**1ère étape : Création du tableau des compteurs C**

Vecteur T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | 7 | 1 | 2 | 4 | 3 |

Vecteur C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 5 | 0 | 1 | 3 | 2 |

**2ème étape : Permutation du plus petit nombre**

Vecteur T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 7 | 5 | 2 | 4 | 3 |

Vecteur C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 5 | 4 | 1 | 3 | 2 |

**3ème étape : Permutation du deuxième plus petit nombre**

Vecteur T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 5 | 7 | 4 | 3 |

Vecteur C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 4 | 5 | 3 | 2 |

**4ème étape : Permutation du troisème plus petit nombre**

Vecteur T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 7 | 4 | 5 |

Vecteur C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 5 | 3 | 4 |

**5ème étape : Permutation du quatrième plus petit nombre**

Vecteur T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 7 | 5 |

Vecteur C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 5 | 4 |

**6ème étape : Permutation du cinquième plus petit nombre avec le plus grand**

Vecteur T

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 7 |

Vecteur C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |

1. But du programme

Ce programme a pour objectif de trier, par ordre croissant, les nombres contenus dans un vecteur. La méthode de tri utilise ici est la méthode de tri par comptage.

1. Principe

Cette méthode se déroule en 2 étapes : la première étape qui permet de créer un second vecteur appelé le tableau des compteurs et la deuxième étape qui utilise le tableau des compteurs pour trier les éléments du vecteur initial par ordre croissant.

Le tableau à trier contient N cases. Le tableau C contient lui aussi N éléments.

Première étape : Elle consiste à créer le vecteur C appelé Tableau des compteurs. Pour ce faire, chaque élément est comparé à tous les autres afin de déterminer le nombre d’éléments qui lui sont inférieurs. Par exemple, on prend le contenu de la première case 5 et on le compare avec tous les autres contenus. On détermine que 4 contenus lui sont inférieurs (1,2,4 et 3). Ce 4 est placé dans la première case du vecteur C. On refait la même chose pour tous les autres éléments. Les résultats de ces N comptages sont rangés dans le vecteur C.

Vecteur C obtenu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 5 | 0 | 1 | 3 | 2 |

*Commentaires* : 4 éléments sont inférieurs à 5, 5 éléments sont inférieurs à 7, 0 élément inférieur à 1, 1 élément est inférieur à 2, 3 éléments sont inférieur à 4 et 2 éléments sont inférieurs à 3.

Deuxième étape : Elle consiste à utiliser le vecteur C appelé Tableau des compteurs pour trier par ordre croissant le vecteur T. Pour ce faire, le premier élément du tableau est permuté avec celui dont le compteur est nul (c’est normal car si il a 0 éléments qui lui sont inférieurs, cela signifie qu’il est le plus petit).

La méthode est la suivante : On cherche 0 dans le vecteur C et on repère le rang de sa case. Ensuite on permute la valeur de la case trouvée avec la case de rang 1. On fait la même chose dans le tableau des compteurs C sinon on perturbe totalement le classement. Puis on cherche 1 dans le vecteur C et on repère le rang de sa case. On permute dans le vecteur T la case 2 avec la case du rang que l’on vient de trouver. On recommence cette opération jusqu’à la valeur N-1.

1. Algorithme

Algorithme "Tri Par Comptage"

Variables indicées : T de N cases de type numérique

C de N cases de type numérique

Variables : numérique : N, I, J, X, Nombre

Début

/\* Saisie de N et contrôle de saisie

Afficher «  Veuillez entrer le nombre de cases du vecteur T (> à 3) »

Saisir N

Tantque N < = 3

Afficher «  Veuillez entrer le nombre de cases du vecteur T > à 3 »

Saisir N

FinTanque

/\* Remplissage du vecteur T qui sera trié par ordre croissant

Pour I allant de 1 à N

Afficher «  Quelle est la valeur de la case de rang », I

Saisir C [I]

Fpour

/\* Affichage du vecteur T avant le tri

Pour I allant de 1 à N

Afficher T [I]

Fpour

/\* Initialisation du tableau des compteurs

Pour I allant de 1 à N

Compteur [I] 🡨 0

Fpour

/\* Création du tableau des compteurs

I 🡨 1

J 🡨 1

Tantque J < = N

I 🡨J + 1

Tantque I <= N

Si T[J] < T [I] Alors C[I] <-- C [I] +1

Sinon C[J] <-- C[J] +1

Fsi

I 🡨 I+1

Ftantque

J🡨 J +1

Ftantque

/\* Affichage du vecteur C précédemment constitué

Pour I allant de 1 à N

Afficher C [I]

Fpour

/\* Utilisation du vecteur C pour trier le vecteur T

Nombre🡨0

J 🡨1

TantQue Nombre < N

/\* Recherche du nombre dans le vecteur C

TantQue c[j] <> Nombre)

J 🡨 J + 1

FinTanque

/\* Permutations du contenu des cases dans T et C

X 🡨 T [Nombre + 1]

T [Nombre + 1] 🡨 T [J]

T [J] 🡨 X

X 🡨 C [Nombre + 1]

C [Nombre + 1] 🡨 C [J]

T [J] 🡨 X

/\* Passage à la permutation suivante

Nombre 🡨 Nombre + 1

J 🡨 Nombre + 1

FinTantque

/\* Affichage du vecteur T après le tri

Pour I allant de 1 à N

Afficher T [I]

Fpour

Fin