

## **SERIE EXERCICES SUR LES TABLEAUX, LES MODULES ET FICHIERS**

### **SERIE EXERCICES SUR LES FICHIERS**

**EXERCICE 1 :** Ecrire un module qui reçoit un tableau d'entiers de 25 cellules. Le module transfère le contenu du tableau dans un fichier d'entiers à organisation séquentielle.

**EXERCICE 2 :** Ecrire un module qui reçoit un fichier d'entiers. Le module détermine le nombre de présence de Val dans le tableau. Val est reçu en donnée.

**EXERCICE 3 :** Ecrire un module qui reçoit un fichier d'entiers. Le module remplace toutes les valeurs égales à VAL par DVAL. Les valeurs de VAL et DVAL sont reçues en paramètre

**EXERCICE 4 :** Ecrire un module qui reçoit un fichier de produits. Le module affiche le contenu du fichier puis détermine le pourcentage de présence de chaque catégorie de produits dans le fichier. Produit (code, catégorie, prix, quantité, catégorie)

**EXERCICE 5 :** Ecrire un module qui reçoit une matrice d'étudiants de 55 lignes et 45 colonnes . Le module transfère le contenu de la matrice dans un fichier d'étudiants. Etudiant (matricule, nom, prénom, classe, filière et moyenne).

**EXERCICE 6 :** Ecrire un module qui reçoit un fichier de caractère puis affiche son contenu.

**EXERCICE 7 :** Ecrire un module qui reçoit un fichier d'étudiants puis permet de rechercher un matricule dans le fichier. Si le matricule est trouvé le sous-programme affiche les données de l'étudiant correspondante sinon il affiche l'étudiant n'est pas dans la liste.

**EXERCICE 8 :** Ecrire un module qui crée un fichier d'entiers. Les valeurs du fichier seront saisies par l'utilisateur. L'arrêt de la saisie est conditionné par la variable réponse.

**EXERCICE 9 :** Ecrire un module qui reçoit le fichier d'entiers crée ci-dessus. Le module crée deux tableaux T1 et T2 contenant respectivement les nombres premiers et les nombres parfaits du fichier.

---

## SERIE EXERCICES SUR LES TABLEAUX, LES MODULES ET FICHIERS

**EXERCICE 10 :** Ecrire un module qui reçoit le même fichier d'entiers puis détermine les valeurs extrêmes du fichier.

**EXERCICE 11 :** Ecrire un module qui reçoit deux fichiers d'entiers et crée le fichier d'intersection.

**EXERCICE 12 :** Ecrire un module qui reçoit deux fichiers d'entiers et crée le fichier de fusion.

**EXERCICE 13 :** Ecrire un module qui reçoit deux fichiers d'entiers et crée le fichier de différence.

**EXERCICE 14 :** Ecrire un module qui reçoit deux fichiers d'entiers triés par ordre croissant, le module crée le fichier d'intersection.

**EXERCICE 15 :** Ecrire un module qui reçoit deux fichiers d'entiers triés par ordre croissant, le module crée le fichier d'union.

**EXERCICE 16 :** Ecrire un module qui reçoit deux fichiers d'entiers triés par ordre croissant, le module crée le fichier de différence.

**EXERCICE 18 :** Ecrire un module qui reçoit deux fichiers de produits, le module crée le fichier de fusion des deux premiers fichiers.

**EXERCICE 19 :** Ecrire un module qui reçoit deux fichiers de produits triés sur le code, le module crée le fichier contenant la synthèse des données des deux premiers fichiers.