Variante 1

Exercice 1 :

Ecrire un algorithme qui demande un nombre à l'utilisateur, et affiche ensuite tous ses diviseurs qui sont inférieurs à 100

Exercice 2 :

Ecrire un algorithme qui pour chaque élément d'un tableau T affiche cet élément si son carré est aussi présent dans le tableau

Exemple : 1 25 5 100 20 10

Eléments dont les carrés sont présents : 1,5,10

Exercice 3 :

a. Ecrire une fonction qui à comme paramètre un entier strictement positif , et qui retourne 1 si ce nombre est parfait et 0 sinon .
Un nombre est dit parfait s'il est égal à la somme de ses diviseurs stricts .
Exemple : 28 = 1+2+4+7+14

b. Ecrire un programme qui affiche la suite de tous les nombres parfaits inférieurs ou égaux à un nombre entier positif donné (noté n)
Voici la liste des nombres parfaits inférieurs à 10000 : 6 , 28 , 496 , 8128

Exercice 4 :

On souhaite écrire un programme permettant de gérer l'ensemble des salariés d'une entreprise. Chaque salarié est connu par : un matricule, un nom, un prénom et un salaire.

a. Créer une Liste (Tableau) T et Saisir un certain nombre de salariés dans le tableau T (chaque salarié est inséré dans le tableau comme un élément de type dictionnaire).

Il faut ensuite :

1. Afficher tous les employés de l'entreprise
2. Supprimer de l'entreprise un salarié dont le matricule est donné par l'utilisateur
3. Ajouter un nouvel salarié, dont les informations sont saisies au clavier, dans le tableau T
4. Sauvegarder les informations de tous les employés dans un fichier '.csv'

Exercice 5 :

Proposer un algorithme, puis un programme qui permet de :

Enregistrer les informations de 10 villes, une ville est caractérisée par un nom, une altitude, les températures de 12 mois.

on affiche les informations de la ville la plus froide au cours de l'année

Variante 2

Exercice 1 :

Un Compte est caractérisé par : un nom, un mot de passe (8 caractères au minimum)

Proposer un algorithme/programme Python qui permet de :

1. Créer une structure Compte
2. Créer une fonction VeriferPW(MotPasse) qui permet de vérifier un mot de passe saisit par l'utilisateur .Elle renvoie 0 si le mot de passe saisit contient moins que 8 caractères et 1 sinon.
3. Créer une fonction CréeCompte(nom,MotPasse) qui crée et qui renvoie une structure compte.
4. Rajouter une procédure AfficherCompte(Compte) qui affiche les informations d'un compte donné.
5. Ecrire un algorithme qui demande de saisir un nom et un mot de passe, vérifier le mot de passe saisit puis si le mot de passe répond bien à la contrainte, il permet de créer un compte et l'afficher. Dans le cas contraire, un message d'erreur s'affiche à l'utilisateur. (Faire appel aux fonctions nécessaires)

Exercice 2 :

Une salle est caractérisée par son code, son libellé, nombreTables et son type. Proposer un programme en python qui permet de :

1. Enregistrer les informations de 20 salles
2. Afficher les informations de la salle qui a le plus petit nombre de tables.

Exercice 3 :

On donne en entrée un tableau de N éléments de type entier. Donner le programme qui affiche les éléments du tableau qui possèdent leurs carrés (t(i)*t(i)) dans le même tableau. Les éléments son rangés dans un ordre aléatoire à l'intérieur du tableau. Exemple : 2 4 7 11 5 9 16 25 . Les éléments dont les carrés sont présents : 1, 2, 4, 5

Tâches :

- Remplissage du tableau
- Recherche des éléments
- Affichage des éléments

Exercice 4 :

Écrire un algorithme qui permet de remplir un tableau T(20).Puis on remplit deux tableaux T1(20) et T2(20) comme suit :

- T1 contient les nombres positifs du tableau T

- T2 contient les nombres négatifs ou nuls du tableau T.

On affiche les Trois tableaux

Tâches :

- Remplissage du tableau T , tableau T1 et tableau T2
- Affichage des 3 tableaux

Exercice 5 :

Ecrire un algorithme qui permet de transférer une matrice M[L,C] en un tableau V à une seule dimension.

afficher M et V

Variante 3

Exercice 1 :

Ecrire un algorithme qui effectue la lecture d'une matrice carrée A ainsi que sa taille n et affiche la trace de A . Afficher la matrice

(Pour une matrice A(ai,j), Trace(A) = ∑ai,i la somme des éléments de la diagonale).

1. remplissage de A
2. Calcule et affichage de la trace
3. Afficher la matrice

Exercice 2:

Rechercher dans une matrice donnée M les éléments qui sont à la fois un maximum sur leur ligne et un minimum sur leur colonne. Ces éléments sont appelés des points-cols.

Exercice 3 :

Ecrire un algorithme puis un programme en python qui permet de :

- lire un entier n
- calculer le produit suivant : P=10* [(1/n)*(1/n-1)*(1/n-2)*(1/n-3)…….*(1/3)*(1/2)*1]
- Afficher le résultat P

Exercice 4 :

Considérons le programme suivant :

```
for i in range(3,20,3):
   if i == 12:
      print(i == 12)
   if i > 15:
      print(i > 15)
```

```
if i < 10:
   print(i < 10)
else:
    print("else")
```

1. Quelle est le résultat affiché si l'utilisateur donne 4 ?
2. Remplacer la boucle for par while.

Exercice 5 :

Vous pouvez proposer un algorithme ou un programme

1. Définir une fonction qui retourne le plus grand entre deux nombres réels
2. Utiliser la fonction déjà définie pour trouver le plus grand nombre entre 10 nombres donnés par l'utilisateur.