

# TP 5

## Consignes pour l'exercice :

- Désactiver Copilot ou tout autre IA, pas d'utilisation d'IA autorisée (si je vous vois l'utiliser c'est 0 au TP).
- Créer un dossier TP4 dans votre répertoire Git du TP1.
- Créer un fichier nommé tp4.c
- Faire un **commit & push** de ce fichier
- Vérifier sur Github que cela a fonctionné.
- Au moins à la fin de chaque étape du TP faire un **commit & push**. Votre **message** de commit doit être **explicite** !!

## Ex1. Gestion des notes

### Objectifs

Suivi de consommation avec emojis, fichiers et fonctions. **Il faudra avoir des fichiers d'en-tête !**

Vous allez écrire un petit programme de suivi de consommation alimentaire.  
L'idée est de suivre chaque jour combien vous consommez de :

1. Eau
2. Café
3. Bonbons
4. Gâteaux
5. Légumes
6. Fruits
7. Protéines

Le programme doit :

- afficher un menu et utiliser des fonctions pour chaque action
- sauvegarder les données dans un fichier avant de quitter
- recharger les données au démarrage
- afficher des emojis en fonction de vos habitudes
- calculer un score de santé simple
- afficher des barres de progression en texte

## Étape 1 : Squelette de base

- Créer un tableau de 7 entiers qui permettra de stocker les consommations des différents éléments.
  - Ecrire une fonction qui affiche le menu
  - Ecrire une fonction qui lit le choix de l'utilisateur
  - Créer dans le main la boucle qui permet d'afficher le menu et de faire le choix, tant que celui-ci n'est pas de quitter.
- 

### Suivi de consommation

1. Ajouter une consommation
2. Afficher le résumé du jour
3. Sauvegarder et quitter

Votre choix : □

## Étape 2 : Consommation

- Créer une fonction *initialiser*, qui permettra d'initialiser le tableau conso.
- Créer une fonction *ajouterConsommation* :
  - o qui affiche les catégories que l'on peut modifier
  - o demande à l'utilisateur laquelle il souhaite changer.
  - o récupère le choix
  - o demande la quantité à modifier
  - o modifie la valeur dans le tableau

Quelle catégorie voulez vous modifier

1. Eau 
2. Cafe 
3. Bonbons 
4. Gateau 
5. Legumes 
6. Fruits 
7. Proteines 

Votre choix : 7

Combien d'unités ajouter ? 20

Consommation mise à jour.

## Étape 3 : Affichage du résumé

- Créer une fonction d'affichage *afficheResume*, qui affichera la consommation faite dans chaque catégorie.

Pour l'instant afficher seulement :

```
===== Resume du jour =====
Eau      :   2
Cafe     :   0
Bonbons  :   0
Gateau   :   0
Legumes  :   3
Fruits   :   0
Proteines:  20
=====
```

#### Étape 4 : Main

Si ce n'est pas déjà fait, mettre à jour le main, pour qu'il utilise, en fonction des choix fait, les fonctions faites précédemment.

#### Étape 5 : Lecture dans un fichier

Pour la sauvegarde, les données seront stockées dans un fichier *consommation.txt*. Il contiendra une unique ligne avec les 7 entiers représentant les consommations, chacune séparée par un espace.

- Créer une fonction *charger*, qui permettra :
  - o Ouvrir *consommation.txt* en lecture
  - o Si l'ouverture rate, elle retournera 0
  - o Lire les 7 entiers et les stocker dans le tableau
  - o Si la lecture échoue à un moment donné : fermer le fichier et retourner 0
  - o Si tout c'est bien passé retourner 1
  - o

#### Étape 6 : Sauvegarde dans un fichier

- Créer une fonction *sauvegarder* :
  - o Ouvrir *consommation.txt* en écriture
  - o Ecrire les 7 entiers séparés par des espaces puis finir par un saut à la ligne
  - o Fermer le fichier
  - o Retourner 1 si tout c'est bien passé, 0 sinon.

## Étape 7: Main

Dans le main :

- au début du programme, appeler la fonction charger, si la fonction retourne 0, afficher « Aucune sauvegarde trouvée » puis appeler la fonction initialiser.
- Appeler la fonction de sauvegarde pour le bon choix. Afficher un message en fonction du succès ou non de la sauvegarde.

## Étape 8 : Emojis selon les quantités

- Modifier la fonction afficheResume pour ajouter les émojis « fixes » de chaque catégorie :

```
===== Resume du jour =====
Eau      : 2 🌊
Cafe     : 0 ☕
Bonbons  : 0 🎂
Gateau   : 0 🍰
Legumes  : 3 🥬
Fruits   : 0 🍎
Proteines: 20 🥐
=====
```

- Ecrire 3 fonctions : humeurBonbons, humeurLegumes, humeurFruits
- humeurBonbons renvoie :
  - o 0 si nbBonbons appartient [0 ; 3]
  - o 1 si nbBonbons appartient [4 ; 7]
  - o 2 si nbBonbons appartient [8 ; 12]
  - o 3 sinon
- Faire pareil pour humeurLegumes et humeurFruits en changeant les valeurs
- Pour ces 3 catégories, ajouter un émojis en plus en fonction de la quantité consommée :
  - Exemple :
    - o pour les bonbons, en plus, afficher un emoji d'humeur en fonction de la valeur renournée par humeurBonbons:
      - 0 → 😊
      - 1 → 😃
      - 2 → 😔
      - 3 → 😈
    - o pour les légumes, utiliser humeurLegumes avec par exemple:
      - 0 → 🥬
      - 1 → 😃
      - 2 → 😄
    - o pour les fruits, utiliser humeurFruits avec par exemple:
      - 0 → 😔
      - 1 → 😃
      - 2 → 😄

## ===== Résumé du jour =====

Eau	:	2	
Café	:	0	
Bonbons	:	0	
Gateau	:	0	
Légumes	:	3	
Fruits	:	0	
Proteines	:	20	

### Étape 9 : Barres de progression

On va représenter les quantités sous forme de barres de 10 cases:

case pleine:

case vide:

- Créer la fonction afficherBarre qui prend en paramètre une valeur et un max :
  - o si max <= 0, utiliser max = 1 pour éviter une division par zéro
  - o borner valeur entre 0 et max
  - o calculer casesPleines = (valeur \* 10) / max
  - o afficher 10 caractères dans une boucle :
    - si i < casesPleines → afficher
    - sinon → afficher
- Modifier afficherRésumé pour appeler afficherBarre après chaque ligne

## ===== Résumé du jour =====

Eau	:	2								
Café	:	0								
Bonbons	:	0								
Gateau	:	0								
Légumes	:	3								
Fruits	:	0								
Proteines	:	20								

### Étape 10 : Objectifs et score de santé

- Dans le main, créer un tableau objectifs de 7 entiers.  
Pour l'eau, la valeur sera de 8, pour les légumes de 5, pour les fruits de 3, pour les protéines de 2. Pour les autres, mettre à 0.  
Ajouter également un nouveau choix au menu :

---

## Suivi de consommation

---

1. Ajouter une consommation
2. Afficher le résumé du jour
3. Afficher les objectifs et le score
4. Sauvegarder et quitter

Votre choix : █

- Faire une fonction pour l'affichage des objectifs et du score.
- L'affichage des objectifs doit donner un résultat semblable :

---

### Objectifs du jour

---

Categorie	Objectif	Atteint ?
-----------	----------	-----------

Eau	:	8
Cafe	:	-
Bonbons	:	-
Gateau	:	-
Legumes	:	5
Fruits	:	3
Proteines	:	2

- Ecrire une fonction calculerScoreSante :
  - o score de base: 50
  - o +10 si eau ≥ objectif eau
  - o +10 si légumes ≥ objectif légumes
  - o +10 si fruits ≥ objectif fruits
  - o +10 si protéines ≥ objectif protéines
  - o si bonbons > 5:
    - enlever 1 point par bonbon au delà de 5
    - malus maximum: 15 points
  - o si café > 3:
    - enlever 2 points par café au delà de 3
    - malus maximum: 20 points
  - o borner le score final entre 0 et 100

Appeler cette fonction depuis afficherObjectifsEtScore et afficher à la suite des objectifs :

Score de santé du jour : 60 / 100

---

- Raccorder au bon choix de menu dans le main.