

TP 4 Fonctions

Consignes pour les 2 exercices :

- 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

Concevoir un programme en C qui gère les notes d'une classe à l'aide de **fonctions clairement séparées**, chacune réalisant **une tâche unique**.

L'objectif principal est de **savoir créer, utiliser et structurer un programme autour de fonctions**, en évitant au maximum du code dans main.

On veut créer un programme qui gère les notes d'une classe de **maximum 30 élèves**, ayant **3 notes chacun**.

Le programme doit fonctionner **uniquement via un menu**, et les calculs (moyennes, max, etc.) doivent être faits **par des fonctions séparées**.

Étape 1 : Fonction afficherMenu

Doit afficher le menu du programme :

- Afficher la liste des options disponibles.
- Ne demande aucune saisie (c'est uniquement de l'affichage).
- Elle ne retourne aucune valeur.

```
===== GESTION DE NOTES =====
1. Saisir le nombre d'eleves
2. Saisir les notes des eleves
3. Afficher toutes les notes
4. Afficher la moyenne d'un eleve
5. Afficher la moyenne generale
6. Afficher la meilleure note de chaque controle
0. Quitter
```

Étape 2 : Fonction lireChoix

Doit lire le choix (menu) saisi par l'utilisateur :

- Lire une valeur depuis le clavier.
- Retourner cet entier au main.

Votre choix : 1

Étape 3 : Fonction saisirNombreEleves

Doit demander à l'utilisateur combien d'élèves sont dans la classe :

- Demander un nombre entre 1 et 30.
- Tant que la valeur est invalide, redemander.
- Retourner la valeur correcte au main.

Entrez le nombre d'eleves (1 a 30) : 31
Valeur invalide.
Entrez le nombre d'eleves (1 a 30) : 2

Étape 4 : Fonction saisirNotes

Doit remplir le tableau de notes :

- Parcourir tous les élèves.
- Pour chaque élève, demander les 3 notes (contrôles).
- Chaque note doit être comprise entre 0 et 20.

Saisie des notes pour 2 eleves et 3 controles.

Eleve 1 :

Note du controle 1 (0 a 20) : 15
Note du controle 2 (0 a 20) : 13
Note du controle 3 (0 a 20) : 11

Eleve 2 :

Note du controle 1 (0 a 20) : 2
Note du controle 2 (0 a 20) : 6
Note du controle 3 (0 a 20) : 20

Étape 5 : Fonction afficherNotes

Doit afficher toutes les notes sous forme de tableau :

- Afficher l'en-tête (Eleve C1 C2 C3)
- Afficher les notes de chaque élève avec leur numéro (1 à *nbEleves*)

Tableau des notes

Eleve	C1	C2	C3
1	15	13	11
2	2	6	20

Étape 6 : Fonction calculerMoyenneEleve

Doit calculer la moyenne d'un élève donné :

- Additionner les 3 notes de l'élève *indiceEleve*.
- Diviser par 3.
- Retourner la moyenne en float.
- L'affichage se fera dans le *main*.

```
Entrez l'indice de l'eleve (1 a 2): 1
Moyenne de l'eleve 1: 13.00
```

Étape 7 : Fonction calculerMoyenneGenerale

Doit calculer la moyenne de la classe :

- Appeler **la fonction précédente** (*calculerMoyenneEleve*) pour chaque élève.
- Faire la moyenne de toutes les moyennes.
- Retourner la moyenne générale.
- Afficher la moyenne dans le *main*.

```
Moyenne generale de la classe: 11.17
```

Étape 8 : Fonction trouverMeilleureNoteControle

Doit renvoyer la meilleure note obtenue pour un contrôle (C1, C2 ou C3) :

- Parcourir tous les élèves.
- Chercher le maximum dans la colonne correspondant au contrôle *indiceControle* (0, 1 ou 2).
- Retourner la meilleure note trouvée.

Étape 9 : Fonction afficherMeilleuresNotes

Doit afficher les meilleures notes des 3 contrôles :

- Appeler *trouverMeilleureNoteControle* trois fois.
- Afficher les résultats sous la forme :

```
Meilleure note au controle 1 : 15
Meilleure note au controle 2 : 13
Meilleure note au controle 3 : 20
```

Ex2. Simulation d'un distributeur de billets

Vous développez un programme pour une banque. Vous devez programmer l'interface d'un distributeur automatique (DAB).

Le DAB ne propose que ces billets : 50, 20, 10 et 5 euros.

Le **montant maximum disponible dans l'appareil est de 1000€**.

L'utilisateur peut retirer plusieurs fois tant qu'il ne quitte pas.

Interdire :

- montant négatif
- montant nul
- montant supérieur à 1000€

Vous devez obligatoirement créer **au minimum** ces fonctions :

Nom	Rôle	Retour	Paramètres
afficher_menu()	Affiche le menu	rien	aucun
saisir_montant()	Lit et retourne un montant	int	aucun
montant_valide(int montant)	Vérifie si le montant est correct	int (0 ou 1)	montant
calcul_distribution(int montant)	Calcule le nombre de billets et les affiche	rien	montant

Etape 1 : Afficher le menu

```
=== BANQUE - DISTRIBUTEUR ===  
1 - Faire un retrait  
2 - Quitter _
```

L'utilisateur retente après chaque opération.

Dans le *main*, demander le choix :

```
Votre choix : 
```

Etape 2 : Demander un montant

Avec la fonction `saisir_montant()`.

```
Montant a retirer : 75
```

Etape 3 : Vérifier la validité

avec `montant_valide(...)`, vérifier que l'on a un multiple de 5 sinon :

```
Montant a retirer : 32
```

```
Erreur : montant invalide.
```

```
Le montant doit etre entre 5 euros et 1000 euros, divisible par 5.
```

Etape 4 : Calculer et afficher les billets

Si valide :

```
Billets distribues :  
1 billets de 50 euros  
1 billets de 20 euros  
1 billets de 5 euros
```

Etape 5 : Boucler tant que l'utilisateur ne quitte pas

Finaliser le *main*.