

**Département des Mathématiques et Informatique**

**Module : Langage de programmation C**

**MIP / S3**

**Mini projet**

**Gestion d’un Restaurant**

**2024-2025**

**Réalisé par :**

* M. EZ-ZAHERY Ahmed Amine
* M. BELAZRI Hamza
* Mme. EL MEHDAOUI Chaimaa
* M. HAMMOU Abderrahmane

**Encadré par :**

* Pr. EL FILALI Sanaa
* Pr. KENDALI Khalid
* Pr. BENAMRI Ichrak
* Mme. MHAMMEDI Sajida
* M. TAZI Mohannad

Remerciement

Nous souhaitons exprimer notre profonde gratitude à Mme Sanaa EL FILALI, professeure du cours de programmation en langage C, pour son enseignement clair, son dévouement et son accompagnement tout au long de ce semestre. Son expertise et sa pédagogie ont été d'une grande aide pour nous permettre de comprendre les fondamentaux et d'acquérir les compétences nécessaires à la réalisation de ce projet.

Nous remercions également les professeurs des travaux dirigés pour leur disponibilité et leur soutien constant. Leur patience et leurs explications détaillées ont grandement contribué à renforcer nos connaissances et à surmonter les difficultés rencontrées lors de notre apprentissage.

Enfin, nous adressons nos sincères remerciements aux encadrants en langage C, dont les conseils précieux et l'encadrement rigoureux ont joué un rôle clé dans la réussite de notre projet. Leur encouragement et leur implication nous ont permis de progresser et de mener ce projet à bien.

Grâce à leur encadrement et à leurs efforts combinés, nous avons pu approfondir nos compétences en programmation et mettre en pratique nos acquis dans le cadre d’un projet concret et enrichissant.

Résumé

Le présent document constitue le fruit du travail réalisé dans le cadre de notre projet de fin de semestre 3, dans le domaine de la programmation en langage C.

Dans le cadre de ce projet, nous avons développé une application de gestion de restaurant. Cette application vise à simplifier la gestion des menus et des commandes en offrant aux utilisateurs une interface claire et intuitive. Grâce à une analyse approfondie des besoins opérationnels d'un restaurant, notre application permet la gestion des plats (ajout, modification, suppression) et des commandes, tout en calculant automatiquement les totaux des commandes effectuées.

L’application repose sur une architecture solide, utilisant des structures et des tableaux pour organiser les données. Elle assure également une navigation fluide entre les différentes fonctionnalités grâce à un menu principal interactif. L’objectif principal de ce projet est d’optimiser la gestion quotidienne des opérations d’un restaurant et d’améliorer l’expérience utilisateur.

L’enjeu de ce projet est de proposer une solution informatique fiable et évolutive, tout en consolidant nos compétences en programmation structurée.

Membre de groupes

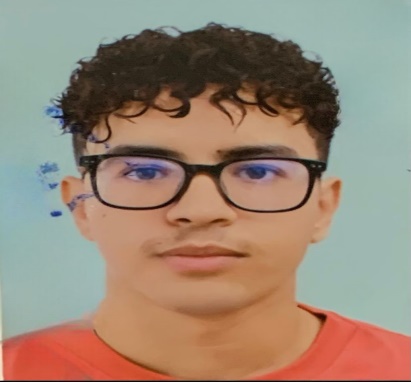
A person with a beard and mustache

Description automatically generated

Mr. EZ-ZAHERY Ahmed Amine

Mr. HAMMOU

Abderrahmane



Mme. EL MEHDAOUI Chaimaa

Mr. BELAZRI Hamza

Sommaire

[Remerciement 2](#_Toc187166743)

[Résumé 3](#_Toc187166744)

[Membre de groupes 3](#_Toc187166745)

[Sommaire 4](#_Toc187166746)

[Introduction 6](#_Toc187166747)

[Chapitre 1 : Fonction Générale 7](#_Toc187166748)

[Introduction : 8](#_Toc187166749)

[Description de la Fonction : 8](#_Toc187166750)

[Corps de la Fonction 9](#_Toc187166751)

[Exemple d'Exécution : 9](#_Toc187166752)

[Conclusion : 10](#_Toc187166753)

[Chapitre 2 : Gestion du Menu 11](#_Toc187166754)

[Introduction : 12](#_Toc187166755)

[Ajouter un plat : 12](#_Toc187166756)

[a) Description de la fonction : 12](#_Toc187166757)

[b) Corps de la fonction : 13](#_Toc187166758)

[Explication : 13](#_Toc187166759)

[c) Exemple d'exécution : 14](#_Toc187166760)

[Modifier un plat : 14](#_Toc187166761)

[a) Description de la fonction : 14](#_Toc187166762)

[b) Corps de la fonction : 15](#_Toc187166763)

[c) Exemple d'exécution : 16](#_Toc187166764)

[Supprimer un plat : 16](#_Toc187166765)

[a) Description de la fonction : 16](#_Toc187166766)

[b) Corps de la fonction : 16](#_Toc187166767)

[Explication : 17](#_Toc187166768)

[c) Exemple d'exécution : 17](#_Toc187166769)

[Chapitre 3 : Prendre une commande 18](#_Toc187166770)

[Introduction : 19](#_Toc187166771)

[Description de la fonction : 19](#_Toc187166772)

[Corps de la fonction : 20](#_Toc187166773)

[Explication : 20](#_Toc187166774)

[Chapitre 4 : Afficher les Commandes du Jour 22](#_Toc187166775)

[Introduction : 23](#_Toc187166776)

[Description de la fonction : 23](#_Toc187166777)

[Corps de la fonction : 24](#_Toc187166778)

[Conclusion Générale 25](#_Toc187166779)

[Bibliographie 26](#_Toc187166780)

# 

Introduction

Dans le cadre de notre projet de fin de semestre 3, notre équipe s'est vue confier la tâche passionnante de concevoir et de développer une application en langage C dédiée à la gestion d’un restaurant. Ce projet vise à simplifier et à optimiser les processus liés à la gestion des plats et des commandes, en fournissant aux utilisateurs un outil fonctionnel et convivial.

Face aux défis opérationnels rencontrés dans les restaurants, tels que la gestion des menus, l’enregistrement des commandes, et le calcul des totaux, il est essentiel de disposer d’une solution informatique efficace. Notre application de gestion de restaurant centralise ces fonctionnalités afin de simplifier la gestion quotidienne et d’améliorer l’expérience utilisateur.

Ce projet repose sur une analyse approfondie des besoins spécifiques des établissements de restauration. Nous avons conçu une application basée sur des structures de données et des techniques de programmation structurée, assurant une interface utilisateur intuitive et des fonctionnalités robustes. L’objectif principal est de fournir une solution fiable et évolutive qui répond aux exigences du secteur de la restauration tout en permettant une utilisation aisée.

Notre application permet aux administrateurs de gérer efficacement les plats en ajoutant, modifiant ou supprimant des éléments du menu. Elle offre également un module pour l’enregistrement des commandes, le calcul automatique des totaux, et l’affichage des commandes effectuées au cours de la journée.

Les fonctions implémentées dans ce projet jouent un rôle central dans la gestion de ces tâches. Chaque fonction a été soigneusement conçue pour répondre à une exigence précise : gestion du menu (ajout, modification, suppression des plats), traitement des commandes (enregistrement, calcul des totaux), et suivi des opérations au jour le jour. Ces fonctions assurent une exécution rapide et fiable des différentes tâches, tout en garantissant la validation des données et en minimisant les risques d'erreurs.

Dans les sections suivantes, nous détaillerons chaque fonction, en présentant son fonctionnement, son corps et des exemples d'exécution, afin de montrer leur contribution essentielle au bon déroulement du système global de gestion du restaurant. Nous démontrerons comment ces fonctions facilitent le travail quotidien des utilisateurs et optimisent la gestion du restaurant.

Chapitre 1 : Fonction Générale

# Introduction :

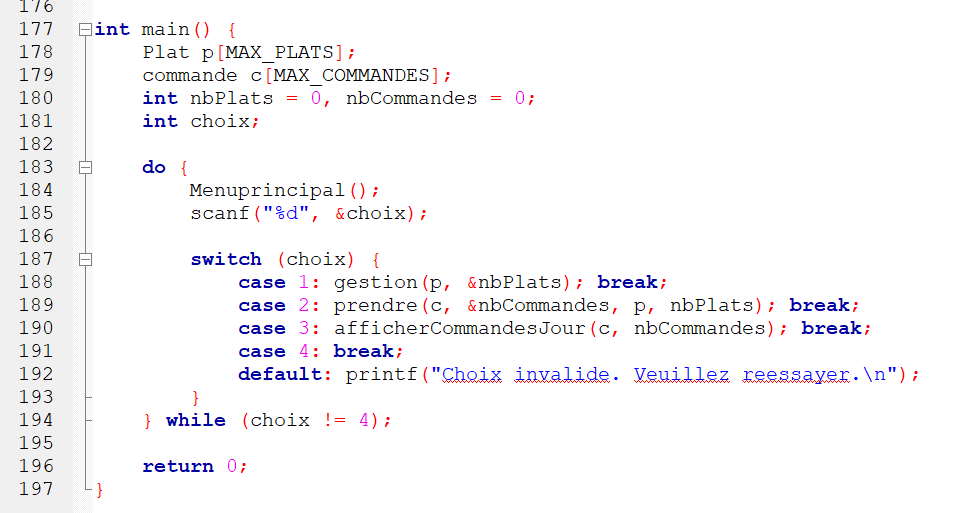
Dans la gestion d'un restaurant, il est crucial d'intégrer toutes les fonctionnalités essentielles dans un flux de travail cohérent et efficace. La fonction main() joue un rôle central en orchestrant les différentes opérations du système, telles que la gestion du menu, la prise de commandes, et l'affichage des commandes du jour. Elle sert d'interface principale, permettant à l'utilisateur d'interagir avec le programme de manière structurée et intuitive. Cette intégration garantit une expérience utilisateur fluide et une gestion centralisée des fonctionnalités, tout en minimisant les erreurs opérationnelles.

# Description de la Fonction :

La fonction main() est le point d'entrée principal du programme. Elle assure le contrôle de l'application en proposant un menu principal à l'utilisateur, dans lequel chaque option correspond à une tâche spécifique :

1. Gestion du menu : Ajouter, modifier, ou supprimer des plats.
2. Prendre une commande : Enregistrer une nouvelle commande en fonction des plats disponibles.
3. Afficher les commandes du jour : Visualiser les commandes enregistrées au cours de la journée.
4. Quitter : Terminer le programme.

Cette fonction repose sur une structure de contrôle en boucle qui permet à l'utilisateur d'exécuter plusieurs actions successives jusqu'à ce qu'il décide de quitter.

Corps de la Fonction **:**

# A black screen with white text Description automatically generatedExemple d'Exécution :

**Étape 1** : L'utilisateur lance le programme, et le menu principal s'affiche :

**Étape 2** :

* Si l'utilisateur choisit 1, la fonction gestion() est appelée pour gérer les plats du menu.
* Si l'utilisateur choisit 2, la fonction prendre() permet de saisir une commande.
* Si l'utilisateur choisit 3, la fonction afficherCommandesJour() affiche les commandes passées.
* Si l'utilisateur choisit 4, le programme se termine.

**Étape 3** : Une fois qu'une action est exécutée, le menu principal réapparaît, et l'utilisateur peut choisir une autre opération ou quitter.

# Conclusion :

La fonction main() est le noyau du programme. Elle relie toutes les fonctionnalités principales et offre une interface utilisateur claire pour interagir avec l'application. Sa structure modulaire et sa boucle de contrôle assurent une gestion fluide et efficace des différentes tâches, répondant ainsi aux besoins opérationnels du système de gestion d'un restaurant.

Chapitre 2 : Gestion du Menu

# Introduction :

Ce chapitre présente les fonctions liées à la gestion du menu du restaurant. Il permet aux administrateurs d'ajouter, modifier, supprimer et afficher les plats disponibles. L'ajout vérifie l'unicité du numéro de plat, la modification permet de mettre à jour le nom ou le prix, et la suppression retire un plat après confirmation. Enfin, l'affichage fournit une vue complète du menu avec les prix. Ces fonctions assurent une gestion fluide et cohérente du menu du restaurant.

# Ajouter un plat :

## Description de la fonction :

La fonction "Ajouter un plat" permet d'ajouter un nouveau plat au menu du restaurant. Elle prend en entrée les informations du plat, à savoir son numéro, son nom et son prix. Avant d'ajouter un plat, la fonction vérifie d'abord si le menu est déjà plein. Si le nombre de plats atteint la capacité maximale (MAX\_PLATS), l'ajout est empêché et un message d'erreur est affiché. Ensuite, elle vérifie si le numéro de plat est unique pour éviter les doublons.

## Corps de la fonction :

### Explication :

1. Vérification du menu plein : La condition if (\*nbPlats >= MAX\_PLATS) vérifie si le nombre de plats a atteint la capacité maximale définie par MAX\_PLATS. Si cette condition est vraie, un message est affiché pour indiquer que le menu est plein et la fonction retourne sans ajouter le plat.
2. Saisie des informations : Si le menu n'est pas plein, la fonction demande à l'utilisateur de saisir le numéro, le nom et le prix du plat.
3. Vérification de l'unicité du numéro de plat : Ensuite, un boucle for parcourt les plats existants pour vérifier si le numéro du plat saisi existe déjà dans le menu. Si oui, la fonction retourne et affiche un message d'erreur.
4. Ajout du plat : Si le numéro est unique, le plat est ajouté et le nombre de plats (nbPlats) est mis à jour.

## A black screen with white text Description automatically generated Exemple d'exécution :

# Modifier un plat :

## Description de la fonction :

La fonction "Modifier un plat" permet à l'administrateur de mettre à jour les informations d'un plat existant dans le menu, telles que le nom et le prix. Avant d'effectuer la modification, la fonction vérifie si le plat avec le numéro spécifié existe dans le menu. Si le plat est trouvé, les nouvelles informations sont saisies et le plat est mis à jour. Sinon, un message d'erreur est affiché.

## Corps de la fonction :

**Explication :**

1. Recherche du plat à modifier : La fonction commence par demander à l'administrateur le numéro du plat à modifier. Ensuite, elle parcourt tous les plats dans le menu pour vérifier si le plat avec ce numéro existe.
2. Modification des informations : Si le plat est trouvé, l'administrateur peut saisir un nouveau nom et un nouveau prix pour le plat. Les anciennes valeurs sont affichées pour référence.
3. Message de succès ou d'erreur : Si le plat a été modifié avec succès, un message est affiché. Si le plat n'existe pas, la fonction affiche un message d'erreur.

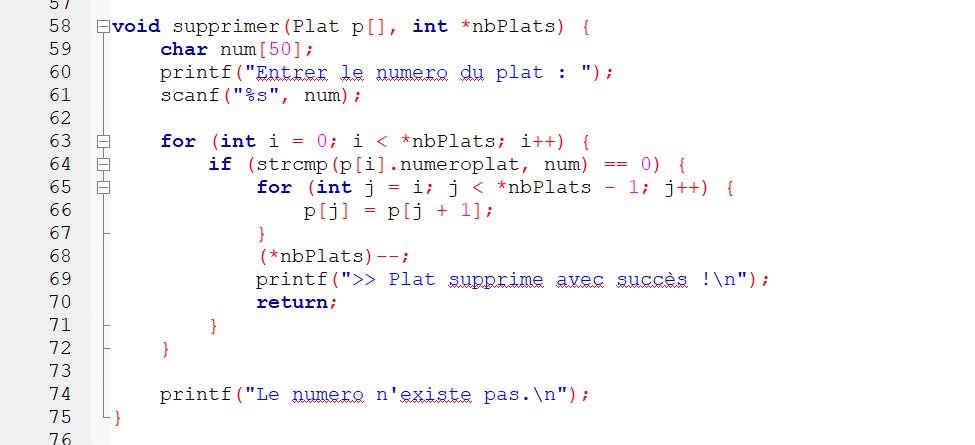
## A black screen with white text Description automatically generatedExemple d'exécution :

# Supprimer un plat :

## Description de la fonction :

La fonction "Supprimer un plat" permet à l'administrateur de retirer un plat du menu en utilisant son numéro. Elle vérifie d'abord si le plat existe dans le menu. Si le plat est trouvé, il est supprimé, et les plats suivants dans le tableau sont décalés pour remplir l'espace laissé. Si le plat n'existe pas, un message d'erreur est affiché.

## Corps de la fonction :

****

### Explication :

1. Recherche du plat à supprimer : La fonction demande à l'administrateur d'entrer le numéro du plat à supprimer. Elle parcourt le tableau des plats pour localiser le plat correspondant.
2. Suppression et décalage : Si le plat est trouvé, il est supprimé en décalant tous les plats suivants d'une position vers le haut pour remplir l'espace laissé.
3. Mise à jour du compteur : Le nombre total de plats est décrémenté pour refléter la suppression.
4. Message de succès ou d'erreur : Un message est affiché pour confirmer la suppression ou signaler que le plat n'a pas été trouvé.

## Exemple d'exécution :

Chapitre 3 : Prendre une commande

# Introduction :

Dans un restaurant, la gestion des commandes constitue un défi majeur. Une mauvaise organisation peut entraîner des erreurs dans les commandes, des retards dans le service ou des clients insatisfaits. Il est donc essentiel de disposer d’un système fiable pour enregistrer les commandes, vérifier la disponibilité des plats et calculer les totaux. La problématique réside dans la nécessité de centraliser ces étapes tout en réduisant les risques d'erreur et en optimisant le temps de traitement, afin d'améliorer l'efficacité opérationnelle et l'expérience client.

# Description de la fonction :

La fonction "Prendre une commande" permet d'enregistrer une commande composée de plusieurs plats. Pour chaque plat commandé, la fonction vérifie s'il existe dans le menu, enregistre la quantité demandée et calcule le total de la commande. Si un plat n'est pas trouvé, un message d'erreur est affiché. Une fois la commande enregistrée, son total est affiché.

# Corps de la fonction :

**A screenshot of a computer program

Description automatically generated**

### Explication :

1. **Vérification du menu :** Si aucun plat n'est disponible dans le menu, la fonction affiche un message et se termine.
2. **Création d'une nouvelle commande :** La commande est initialisée avec un numéro unique et un total de 0.
3. **Enregistrement des plats dans la commande :**
   * L'utilisateur entre le numéro du plat pour chaque élément de la commande.
   * La fonction recherche le plat dans le menu. Si trouvé, elle demande la quantité, calcule le coût total pour ce plat et l'ajoute au total général de la commande.
   * Si le plat n'est pas trouvé, un message d'erreur est affiché.
4. **Finalisation de la commande :** La commande est ajoutée à la liste des commandes, et son total est affiché.

Chapitre 4 : Afficher les Commandes du Jour

# Introduction :

Dans un restaurant, le suivi des commandes effectuées au cours de la journée est crucial pour assurer une gestion efficace des opérations. Cependant, sans un système structuré, il peut être difficile de retrouver rapidement les détails des commandes, d'évaluer les performances quotidiennes ou d'identifier les éventuelles anomalies. La problématique réside dans la nécessité de fournir un moyen simple et organisé pour afficher et consulter les commandes du jour, tout en garantissant une présentation claire et détaillée pour faciliter l'analyse et la prise de décisions.

# Description de la fonction :

La fonction "Afficher les commandes du jour" permet de présenter toutes les commandes enregistrées pendant la journée. Elle affiche pour chaque commande son numéro, les détails du plat commandé, la quantité, et le total associé. Si aucune commande n’a été effectuée, un message indique qu’il n’y a pas de données à afficher.

# Corps de la fonction :

**A screenshot of a computer code

Description automatically generated**

**Explication :**

1. **Vérification des commandes :**
   * Si aucune commande n’a été enregistrée (nbCommandes == 0), la fonction affiche un message informant qu’il n’y a aucune commande.
   * La fonction se termine immédiatement dans ce cas.
2. **Affichage des commandes :**
   * Pour chaque commande enregistrée dans le tableau c[], la fonction affiche les informations suivantes :
     + Le numéro de la commande (numerocommande).
     + Le numéro du plat associé (numeroplat).
     + La quantité commandée (quantite).
     + Le total de la commande en devise MAD (total).
   * Une séparation est affichée entre chaque commande pour plus de lisibilité

Conclusion Générale

Le développement de notre application de gestion de restaurant en langage C nous a permis d’aborder plusieurs problématiques essentielles liées à l’organisation et à la gestion des activités quotidiennes d’un restaurant. Grâce à l’utilisation de structures de données adaptées et de fonctions spécifiques, nous avons conçu un outil capable de répondre efficacement aux besoins opérationnels tels que la gestion des plats, l’enregistrement des commandes, et l’affichage des commandes du jour.

Chaque fonction développée joue un rôle clé dans l’automatisation des processus : de la gestion du menu à la prise de commandes, en passant par l’affichage des informations nécessaires pour un suivi optimal. Ces fonctions ont été conçues pour garantir une interaction intuitive avec l’utilisateur tout en minimisant les erreurs, comme la vérification de l’unicité des plats ou l’affichage clair des données.

Les problématiques abordées dans ce projet, telles que la centralisation des données, la réduction des erreurs humaines et l'amélioration de la lisibilité des informations, ont trouvé des solutions efficaces grâce à une approche méthodique et structurée. L’utilisation des structures comme les tableaux et les champs struct ont permis de manipuler les données de manière organisée et performante.

En conclusion, ce projet met en lumière la puissance du langage C pour résoudre des problématiques concrètes et complexes, tout en offrant une expérience utilisateur pratique et fonctionnelle. Il constitue une base solide pour de futures améliorations, telles que l’intégration d’une interface graphique ou le stockage des données dans des fichiers, afin de rendre l’application encore plus robuste et polyvalente.

Bibliographie

* Stack Overflow. *Questions et Réponses sur le Langage C*. Disponible sur : <https://stackoverflow.com>
* w3schools. *C Language Reference*. Disponible sur :

<https://www.w3schools.com>