

# Introduction

Le programme de gestion d'agenda est conçu pour permettre aux utilisateurs de gérer leurs événements. Il offre des fonctionnalités pour ajouter, supprimer, afficher, sauvegarder et charger des événements, tout en maintenant une structure de données organisée et facile à manipuler. Les structures de données utilisées, telles que Evenement, Nodes et Agenda, ainsi que les fonctions associées, permettent une gestion dynamique et flexible des événements. Ce document fournit une vue d'ensemble des structures et des fonctions clés du programme, accompagnée de brèves explications pour chaque élément.

## A) Partie algorithmique

### 1) Structure utilisée

**Evenement** : Cette structure représente un événement avec des attributs tels que le titre, la date (jour, mois, année), l'heure (heure, minute) et un commentaire. Elle est utilisée pour stocker les détails de chaque événement dans l'agenda.

**Nodes** : Cette structure représente un nœud dans une liste chaînée d'événements. Chaque nœud contient un événement et un pointeur vers le nœud suivant, permettant ainsi de créer une liste chaînée d'événements.

**Agenda** : Cette structure représente un agenda, qui est essentiellement une liste chaînée de nœuds. Elle contient un pointeur vers le premier nœud de la liste, permettant de gérer une collection d'événements de manière dynamique.

### 2) Fonction utilisée

**ajouterEvenement(Agenda, Evenement)\*** : Cette fonction ajoute un nouvel événement à l'agenda en insérant un nœud dans la liste chaînée triée par date et heure.

**comparerEvenements(Evenement, Evenement): int\*\*** : Cette fonction compare deux événements en fonction de leur date et heure, retournant un entier indiquant si le premier événement est avant, après ou égal au second.

**chargerAgenda(Agenda, const char)\*\*** : Cette fonction charge les événements depuis un fichier spécifié et les ajoute à l'agenda.

**joursDansMois(int, int): int** : Cette fonction calcule le nombre de jours dans un mois donné, en tenant compte des années bissextiles pour février.

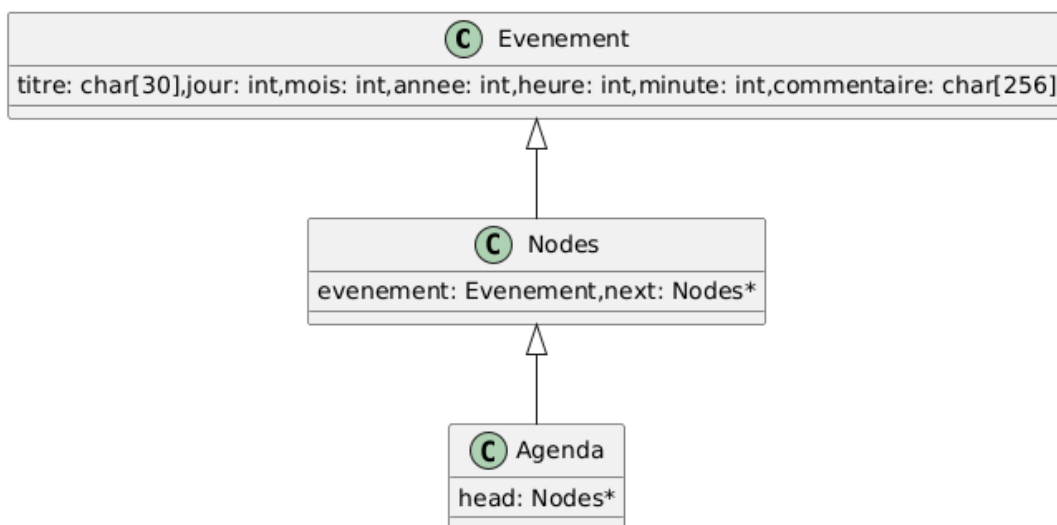
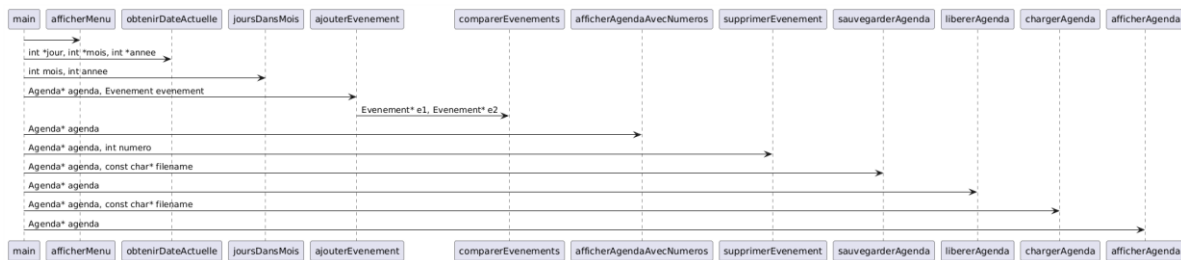
**libererAgenda(Agenda)\*** : Cette fonction libère toute la mémoire allouée pour les nœuds de l'agenda, réinitialisant ainsi l'agenda.

**sauvegarderAgenda(Agenda, const char)\*\*** : Cette fonction sauvegarde tous les événements de l'agenda dans un fichier spécifié, en formatant chaque événement de manière lisible.

**supprimerEvenement(Agenda, int)\*** : Cette fonction supprime un événement spécifique de l'agenda en fonction de son numéro, en parcourant la liste chaînée et en ajustant les pointeurs.

**obtenirDateActuelle(int, int, int\*)\*\*** : Cette fonction récupère la date actuelle du système et la stocke dans les variables passées en paramètres pour le jour, le mois et l'année.

`afficherAgenda(Agenda)*` : Cette fonction affiche tous les événements de l'agenda dans un format lisible, sans numérotation.



- Cas avec jours inférieurs à 0 :

- Cas avec mois supérieurs à 12 : `Entrez la date (JJ/MM/AAAA): 12/50/2050`  
`Date invalide. Veuillez réessayer.`
- Cas avec mois inférieurs à 0 : `Entrez la date (JJ/MM/AAAA): 12/-1/2050`  
`Date invalide. Veuillez réessayer.`
- Cas avec année inférieure à 2025 : `Entrez la date (JJ/MM/AAAA): 12/12/2001`  
`Date invalide. Veuillez réessayer.`
- Cas avec année supérieure à 2100 : `Entrez la date (JJ/MM/AAAA): 12/12/2200`  
`Date invalide. Veuillez réessayer.`
- Cas avec un char : `Entrez la date (JJ/MM/AAAA): t`  
`Format de date invalide. Veuillez réessayer.`
- Cas avec une date valide mais passée : `Entrez la date (JJ/MM/AAAA): 22/01/2025`  
`La date est déjà révolue. Veuillez entrer une date future.`

Bon jeu de donné :

```
Entrez la date (JJ/MM/AAAA): 12/12/2050
Entrez l'heure (HH/MM): 
```

#### b) Entrer l'heure

Les données à intégrer doivent être au format HH/MM. HH doit être un entier entre 0 et 23 et MM doit être un entier entre 0 et 59. Ci-dessous, des jeux de données qui ne fonctionnent pas :

- Cas avec heure inférieur à 0 : `Entrez l'heure (HH/MM): -5/12`  
`Heure ou minute invalide. Veuillez réessayer.`
- Cas avec heure supérieur à 23 : `Entrez l'heure (HH/MM): 24/45`  
`Heure ou minute invalide. Veuillez réessayer.`
- Cas avec minute inférieur à 0 : `Entrez l'heure (HH/MM): 12/-5`  
`Heure ou minute invalide. Veuillez réessayer.`
- Cas avec minute supérieur à 59 : `Entrez l'heure (HH/MM): 12/60`  
`Heure ou minute invalide. Veuillez réessayer.`
- Cas avec un char : `Entrez l'heure (HH/MM): test`  
`Heure ou minute invalide. Veuillez réessayer.`

Bon jeu de donné :

```
Heure ou minute invalide. Veuillez réessayer.
Entrez l'heure (HH/MM): 12/35
```

#### c) Entrer le titre

Le choix du titre est libre dans la limite de 30 char. Un nouveau titre sera demandé si la taille est trop grande.

```
Entrez le titre (max 30 caractères): sssssssssssssssssssssssssssssssssssssssssssssssssssssssss
Le titre est trop long. Veuillez entrer un titre de 30 caractères maximum.
Entrez le titre (max 30 caractères): sss
Entrez le commentaire: 23
```

#### d) Entrer le commentaire

Le choix du commentaire est libre dans une limite de 256 char. Un nouveau commentaire sera demandé si la taille est trop grande.

```
Entrez le commentaire: ddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
ddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
ddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
Le commentaire est trop long. Veuillez entrer un commentaire de 256 caractères maximum.
Entrez le commentaire: d
```

#### 2) Supprimer un évènement

L'ID de l'évènement à supprimer est requis. Un mauvais choix ramène au menu.

```
Entrez le numero de l'evenement a supprimer: 6
Numero d'evenement invalide.
```

```
Entrez le numero de l'evenement a supprimer: a
Entrée invalide. Veuillez entrer un numéro valide.
```

#### 3) Sauvegarder un agenda

La fonction de sauvegarde permet d'enregistrer son fichier sous n'importe quel format. Le format à préférer est le format .txt. Un fichier déjà existant verra ses données entièrement écrasées.

- Exemple d'enregistrement en .json

```
Choisissez une option: 3
Entrez le nom du fichier: est.json
```

- Format attendu

```
Choisissez une option: 3
Entrez le nom du fichier: agenda.txt
```

#### 4) Charger un agenda

Cette fonction permet d'ouvrir un agenda préalablement enregistré par notre programme ou

respectant les conventions d'écriture.

```
Choisissez une option: 4
Entrez le nom du fichier: agenda.txt
```

#### 5) Menu

Le menu nous demande un int entre 1 et 6. Un char ou un mauvais nombre sont de mauvaises réponses.

```
Choisissez une option: d
Entrée invalide. Veuillez entrer un numéro valide.
```

```
Choisissez une option: 45
Option invalide.
```

## Conclusion

En résumé, le programme de gestion d'agenda est une solution pour la gestion des événements. Les structures `Evenement`, `Nodes` et `Agenda` forment la base de la gestion des données, tandis que les fonctions telles que `ajouterEvenement`, `supprimerEvenement`, `sauvegarderAgenda` et `chargerAgenda` offrent des fonctionnalités essentielles pour manipuler ces données. Grâce à cette organisation, le programme permet aux utilisateurs de gérer leurs événements de manière efficace.