

# Documentation Mini Projet Langage C

Caculli Giorgio & Jędrzej Tyranowski

12 décembre 2020

## **Résumé**

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Le langage C . . . . .	3
1.2	Fonctions générales utilisées . . . . .	3
1.2.1	Qu'est-ce l'allocation de mémoire dynamique? . . . . .	3
1.2.2	Qu'est-ce <code>malloc()</code> . . . . .	3
1.2.3	Qu'est-ce <code>calloc()</code> . . . . .	3
1.2.4	Pourquoi utiliser <code>calloc()</code> . . . . .	3
1.2.5	Pourquoi utiliser <code>malloc()</code> . . . . .	3
1.2.6	Syntaxe de <code>calloc()</code> . . . . .	3
1.2.7	Exemples de <code>calloc()</code> . . . . .	3
1.2.8	Syntaxe de <code>malloc()</code> . . . . .	3
1.2.9	Exemples de <code>malloc()</code> . . . . .	3
1.2.10	Differences entre <code>calloc()</code> et <code>malloc()</code> . . . . .	3
<b>2</b>	<b>Listes chaînées</b>	<b>4</b>
2.1	Création d'un nouveau nœud . . . . .	4
2.2	Insertion d'un nœud dans une liste chaînée . . . . .	4
2.3	Suppression d'un nœud d'une liste chaînée . . . . .	4
2.4	Affichage d'une liste chaînée . . . . .	4
<b>3</b>	<b>Énoncé</b>	<b>5</b>
<b>4</b>	<b>Programme</b>	<b>6</b>
4.1	Mode d'emploi . . . . .	6
<b>5</b>	<b>Code</b>	<b>7</b>
5.1	Structures . . . . .	7
5.2	Fonctions . . . . .	8

# 1 Introduction

## 1.1 Le langage C

## 1.2 Fonctions générales utilisées

### 1.2.1 Qu'est-ce l'allocation de mémoire dynamique ?

### 1.2.2 Qu'est-ce malloc()

### 1.2.3 Qu'est-ce calloc()

### 1.2.4 Pourquoi utiliser calloc()

### 1.2.5 Pourquoi utiliser malloc()

### 1.2.6 Syntaxe de calloc()

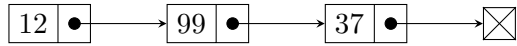
### 1.2.7 Exemples de calloc()

### 1.2.8 Syntaxe de malloc()

### 1.2.9 Exemples de malloc()

### 1.2.10 Differences entre calloc() et malloc()

## 2 Listes chaînées



### 2.1 Création d'un nouveau nœud

### 2.2 Insertion d'un nœud dans une liste chaînée

### 2.3 Suppression d'un nœud d'une liste chaînée

### 2.4 Affichage d'une liste chaînée

### 3 Énoncé

## 4 Programme

### 4.1 Mode d'emploi

## 5 Code

### 5.1 Structures

```
1 typedef struct personne
2 {
3     int id;
4     char nom[25];
5     char prenom[25];
6     int formateur;
7     int nb_formation;
8     int formations[30];
9     int nb_jours_indisponible;
10    int jours_indisponible[7];
11    int reduction;
12    int val_reduction;
13 } personne;

1 typedef struct noeud_db_personne
2 {
3     personne *p;
4     struct noeud_db_personne *next;
5 } noeud_db_personne;

1 typedef struct db_personne
2 {
3     noeud_db_personne *head;
4 } db_personne;

1 typedef struct noeud_formation
2 {
3     personne *p;
4     struct noeud_formation *next;
5 } noeud_formation;

1 typedef struct formation
2 {
3     int id;
4     char nom[40];
5     float prix;
6     int nb_jours;
7     int jours[7];
8     float heures[24];
9     float durees[10];
10    int nb_prerequis;
11    int prerequis[10];
12    noeud_formation *head;
13 } formation;

1 typedef struct noeud_db_formation
2 {
3     formation *f;
4     struct noeud_db_formation *next;
5 } noeud_db_formation;

1 typedef struct db_formation
2 {
3     noeud_db_formation *head;
4 } db_formation;
```

## 5.2 Fonctions

```
1 personne *creer_personne( char nom[], char prenom[], int formateur );

1 void afficher_personne( personne *p );

1 db_personne *creer_db_personne();

1 void ajouter_db_personne( db_personne *db, personne *p );

1 int supprimer_db_personne( db_personne *dbp, int id );

1 void afficher_db_personne( db_personne *db );

1 personne *get_personne( db_personne *db, char nom[], char prenom[], int formateur );

1 formation *creer_formation( char nom[], float prix );

1 int ajouter_formation( formation *f, personne *p );

1 int supprimer_personne_de_formation( formation *f, int id );

1 void afficher_formation( formation *f );

1 db_formation *creer_db_formation();

1 void ajouter_db_formation( db_formation *db, formation *f );

1 int supprimer_db_formation( db_formation *dbf, int id );

1 formation *get_formation( db_formation *dbf, char nom_formation[] );

1 void afficher_db_formation( db_formation *dbf );

1 void menu_creer_formation( db_formation *f );

1 void menu_creer_personne( db_personne *p );

1 int menu_creer( db_formation *f, db_personne *p );

1 void menu_ajouter_formation( db_formation *f, db_personne *p );

1 void menu_supprimer_personne( db_formation *dbf, db_personne *dbp );

1 void menu_supprimer_formation( db_formation *dbf, db_personne *dbp );

1 int menu_supprimer_personne_de_formation( db_formation *dbf );

1 int menu_supprimer( db_formation *dbf, db_personne *dbp );

1 int menu_affichage( db_formation *f, db_personne *p );

1 int menu( db_formation *f, db_personne *p );

1 int main( void );
```