
LANGUAGE C

ESGI

CHALLENGE

Objectifs :

Ce TP est divisé en 2 parties :

1. Prise en main de l'environnement de développement : Editeurs de textes, gcc et devC++. A la fin de cette partie, vous serez capable :
 - i. Produire un exécutable avec gcc à partir de fichiers sources
 - ii. Lancer le fichier exécutable obtenu grâce à gcc.
 - iii. Créer un projet dans DevC++
2. Travail en binôme avec compte-rendu (Lien vers Git).

Partie 1 : Edition, compilation et exécution de votre premier programme en C (30 minutes)

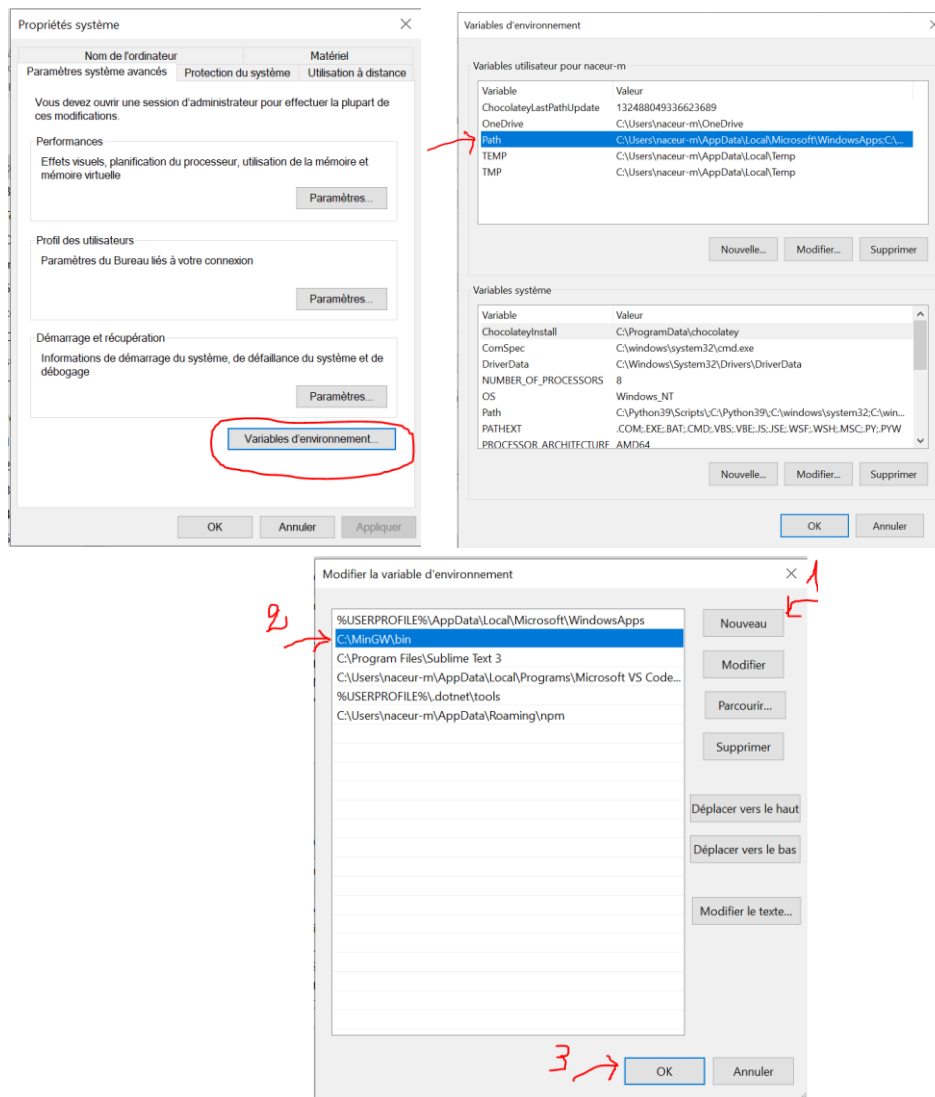
Pour ce TP, nous avons choisi de vous faire manipuler les outils de base les plus couramment utilisés dans le développement de programme C :

- Le terminal pour l'exécution des commandes
- L'outil de compilation gcc
- Un éditeur de code C, ici on utilisera Sublime Text (Sauf si vous n'arrivez pas à installer Sublime Text ou si vous avez une préférence pour un autre éditeur de texte)
- Un environnement de travail : DevC++

Installation des outils nécessaires

- 1- **Installation du GCC** : la notice se trouve sur madoc

Une fois l'installation terminée et si vous êtes sous windows, il faut ajouter à la variable d'environnement : path C:\mingw\bin. Pour cela, allez dans : Menu démarrer → rechercher : Modifier les variables d'environnement système



Afin de vérifier que gcc est bien installé, ouvrez le terminal (win+ r et tapez cmd). Votre terminal est ouvert, tapez alors **gcc --version**, si vous avez ce message, c'est que gcc est bien installé et a bien été ajouté au path.

```
C:\Users\naceur-m>gcc --version
gcc (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

2- Installation de Sublime Text : <https://www.sublimetext.com/3>

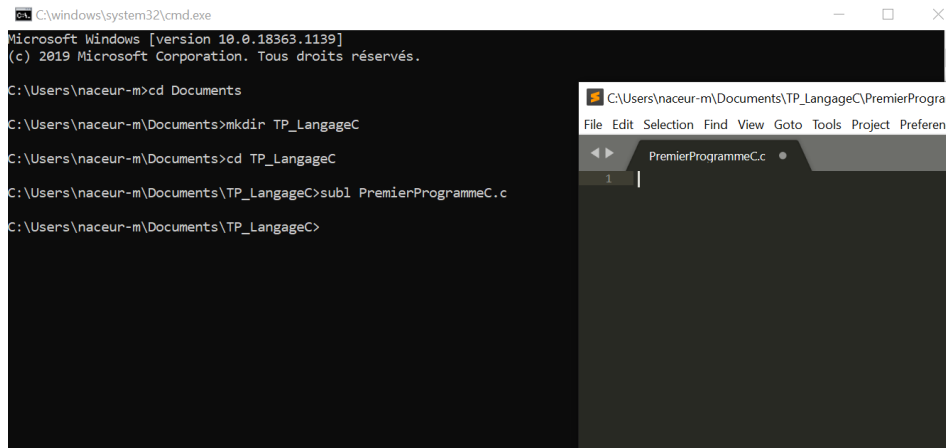
Après installation, vous pouvez ouvrir Sublime Text de différentes manières :

- Si vous êtes déjà dans le terminal, tapez **subl** (avant, il faut l'ajouter au path)
- Sinon, directement en cliquant sur l'icône

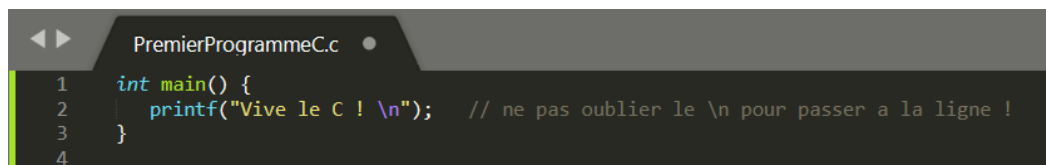
Votre premier exécutable avec gcc

- 1- Ouvrez une fenêtre « **Terminal** » (dans la barre de recherche, tapez cmd)
- 2- Accédez à votre répertoire de travail en tapant : **cd chemin_d_'accès**

- 3- Créez un répertoire de travail pour les fichiers de ce TP en tapant : **mkdir TP1_LangageC** et allez dans ce répertoire en tapant : **cd TP_LangageC**
- 4- Créez et ouvrez un fichier PremierProgrammeC.c en tapant : **subl PremierProgrammeC.c** (attention sous windows, la commande subl ne fonctionne que si Sublime Text a été ajouté au path) ou directement en ouvrant Sublime Text et en créant un nouveau fichier (n'oubliez pas de l'enregistrer dans votre répertoire de travail).



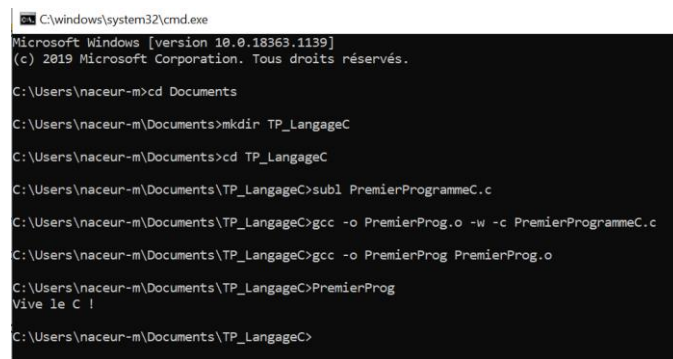
- 5- Le fichier PremierProgrammeC.c est ouvert dans sublime Text. Tapez les lignes suivantes :



- 6- Lancez la compilation. Pour cela, il faut revenir dans le terminal et au niveau du répertoire créé (qui est déjà ouvert, sinon vous devez vous déplacer dans le répertoire TP1_LangageC) et tapez la commande suivante : **gcc -o PremierProg.o -w -c PremierProgrammeC.c**

- a. Que remarquez-vous ?
- b. Que faut-il ajouter et pourquoi ?

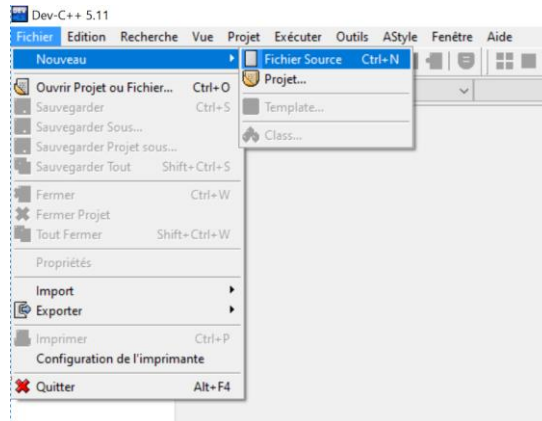
- 6- Reprenez le fichier PremierProgrammeC.c et ajoutez-y les lignes manquantes puis relancer la compilation. S'il n'y a plus de warning, vous pouvez générer votre premier exécutable en tapant : **gcc -o PremierProg PremierProg.o** puis en tapant directement **PremierProg**. Vous devez voir afficher « Vive le C ! ».



Vos premiers pas avec DevC++

Pour l'installation de DevC++ sur vos machines personnelles, vous pouvez commencer par le télécharger [ici](#).

- 1- Lancez DevC++
- 2- Ouvrez un nouveau fichier source en cliquant sur Fichier → Nouveau → Fichier source ou (CTRL+N)



- 3- Tapez le programme suivant :

```
#include<stdio.h>

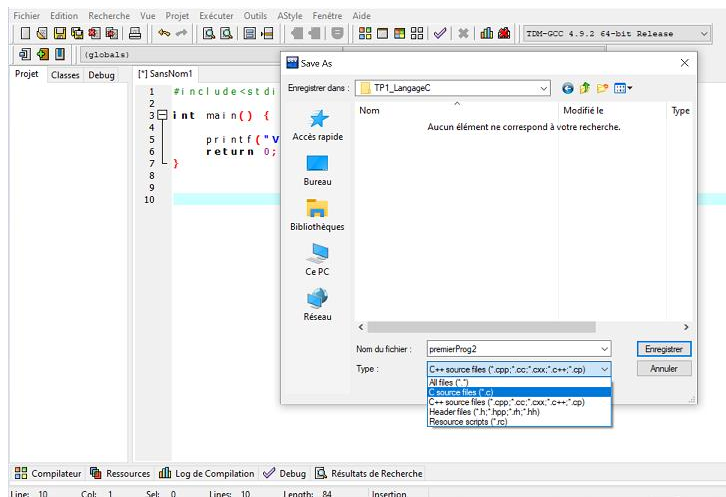
int main() {

    printf("Vive le C ! ");

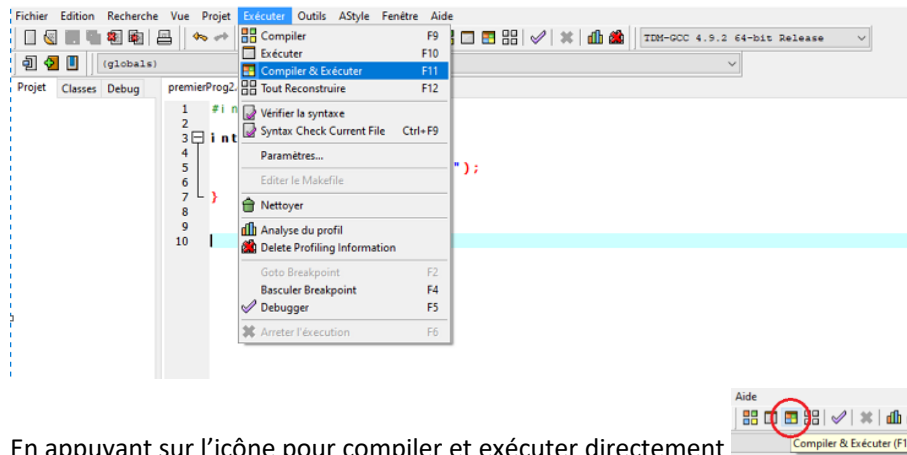
    return 0;

}
```

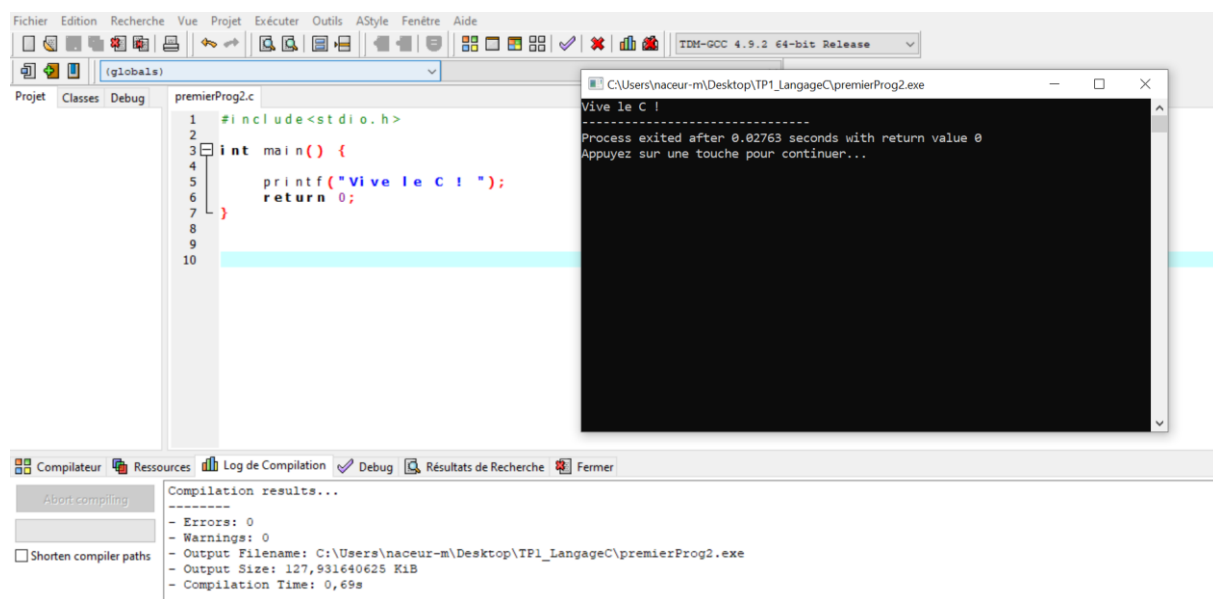
- 4- Enregistrez-le sous votre répertoire de travail premierProg2.c. **Remarque** : attention veillez à bien choisir c et non pas cpp



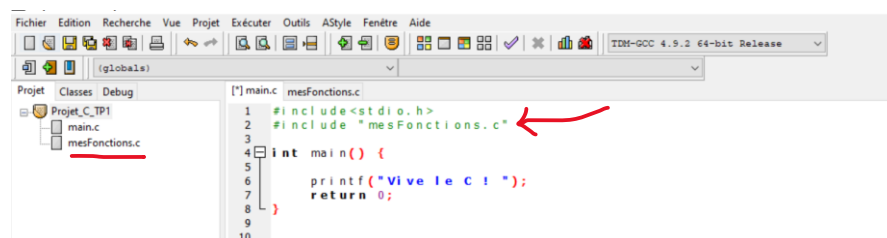
- 5- Nous allons, compiler et exécuter le programme que nous venons d'écrire. Pour cela il y a plusieurs façons de faire :
 - a. En tapant F9, puis F10 ou directement F11
 - b. En cliquant sur l'onglet Exécuter → Compiler ensuite Exécuter → Exécuter (ou directement Exécuter -> Compiler & Exécuter)



- c. En appuyant sur l'icône pour compiler et exécuter directement
- 6- La compilation indique qu'il n'y a pas eu d'erreurs de compilation, ni de warning et une fenêtre contenant le résultat de l'exécution s'ouvre.



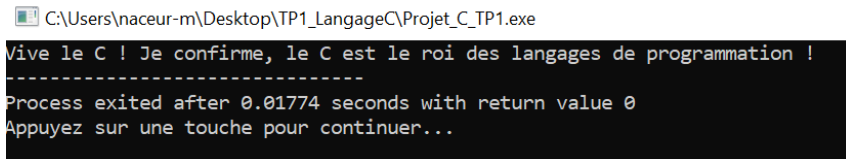
- 7- Nous pouvons également créer un projet et créer autant de fichiers sources qu'on veut. Pour cela, il suffit d'aller dans Fichier → Nouveau Projet et suivre l'assistant de création de nouveau projet en sélectionnant le langage C et en choisissant « Console Application » comme type de projet.
- Créez un nouveau projet en C intitulé «Projet_C_TP1». Par défaut, le fichier main.c est créé, vous pouvez effacer le contenu généré et tapez votre code. Ici on va reprendre le code du programme « Vive le C ! ». Testez l'exécution de votre projet (il vous demandera de confirmer que c'est bien le main.c qu'il faut exécuter comme fichier principal)
 - Créez un nouveau fichier dans le projet en faisant un clic droit sur le nom du projet et en sélectionnant nouveau fichier et enregistrez-le sous le nom « mesFonctions.c » puis ajoutez dans le fichier « main.c » au niveau des bibliothèque le nom du fichier créé en tapant : #include « mesFonctions.c ».



- c. Dans le fichier « mesFonctions.c, tapez le code de la fonction suivante (il s'agit ici d'une fonction sans retour)

```
#include <stdio.h>
void confirmation(){
    printf("Je confirme, le C est le roi des langages de programmation !
}
```

- d. Il suffit de relancer la compilation et l'exécution et vous verrez apparaître les deux messages.



```
C:\Users\naceur-m\Desktop\TP1_LangageC\Projet_C_TP1.exe
Vive le C ! Je confirme, le C est le roi des langages de programmation !
-----
Process exited after 0.01774 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Remarque : Avec d'autres outils, il faut include le fichier « mesFonctions.c » dans le fichier principal « main.c » et cela en ajoutant la ligne «#include "mesFonctions.c"».

Ce qui est demandé !

Il s'agit ici de travailler avec les différentes notions vues en cours : structures de données et tableaux de structures de données en mettant en pratique vos acquis concernant le raisonnement algorithmique.

- 1- Le travail se fera en binôme.
- 2- Ce travail comportera deux parties mais un seul compte-rendu est demandé.
- 3- **Rendu :** un dossier compressé contenant deux sous-dossiers
- 4- **Format du rendu :** lien vers votre Git

Partie 1 : Ma première structure de données ! (2 heures)

Un caractère est représenté par son Code ASCII. La structure ci-dessous permet de représenter les lettres ainsi que les codes ASCII correspondants :

```
typedef struct ASCII_Letter{
    char letter ;
    int code ;
};
```

- 1- Écrire une fonction qui demande à un utilisateur de saisir un caractère et construit la structure correspondante. Exemple : l'utilisateur saisit 'a' → la fonction renvoie une structure ASCII_lettre avec les champs : 'a' et 97.
- 2- Écrire une fonction qui initialise un tableau de n structures struct ASCII_letter.
- 3- Écrire une fonction qui à partir d'une chaîne caractères saisie par un utilisateur remplit un tableau de structures déjà initialisé (alloué).

- 4- Définir une nouvelle structure `struct codage1` qui enregistre deux caractères. Le caractère et le caractère qui se situe à une distance égale à `n` (en utilisant la fonction écrite en 1-2).
- 5- Écrire une fonction qui à partir d'une chaîne caractères saisie par un utilisateur remplit un tableau de `n` structures `struct codage1`.
- 6- Écrire une fonction qui permet de renvoyer le nombre de fois qu'une structure apparaît dans un tableau de type `struct codage1`.

Remarque : N'oubliez pas de tester vos fonctions dans un programme principal. Les fonctions de saisie et d'affichage seront implémentées indépendamment du programme principal.

Partie 2 : Mes mangas préférés ! (2 heures)

Nous allons travailler sur l'énoncé donné dans le cours : Mon répertoire MANGA enrichi avec quelques fonctions en plus. Pour cela, vous travaillerez en binôme. Un nouveau projet MonRepertoireMANGA sera créé.

On se propose d'implémenter une application permettant de gérer des données relatives à nos MANGA préférés. Nous nous intéressons à seulement trois données (titre, âge requis et prix).

Guides Mangas (les) - Bases de données



Vous êtes au début du projet et vous allez implémenter un code en C qui permet de créer une structure de données permettant de définir un MANGA. Un Manga est représenté par son titre (une chaîne de caractères de 50 caractères maximum), l'âge requis pour l'acheter et le prix affiché au public.

La structure de donnée définissant un MANGA est la suivante :

```
struct structure_manga{
    char Titre[50];
    int age;
    float prix;
};

typedef struct structure_manga MANGA;
```

Une fois votre structure créée, vous devez écrire les fonctions suivantes :

- 1- Une fonction qui permet de créer une structure avec les données saisies par l'utilisateur. Cette fonction retourne l'adresse de la structure créée.
- 2- Une fonction qui permet d'afficher les informations d'un MANGA donné en paramètres

- 3- Une fonction qui permet à un utilisateur d'ajouter un nombre fini (n) de ses MANGA préférés. Pour cela, vous devez créer dynamiquement un tableau contenant les MANGA qu'il souhaite ajouter et retourner son adresse (pointeur vers le tableau). Deux variantes de cette fonction sont demandées :
 - a. Le tableau contient des structures MANGA
 - b. Le tableau contient les adresses des structures MANGA
- 4- Une fonction « menu », donnant la possibilité à un utilisateur de :
 - a. Afficher les informations de tous les MANGA stockés dans le tableau
 - b. Afficher les informations de tous les MANGA qui sont accessibles à partir d'un certain âge donné en paramètre.
 - c. Une fonction qui permet de compter le nombre de MANGA ayant un titre qui commence par une lettre donnée en paramètre
 - d. Une fonction qui permet de chercher un MANGA en donnant en paramètres le tableau de structure et le titre du MANGA. La fonction doit renvoyer le prix du MANGA donné.
 - e. Une fonction qui renvoie le prix total (somme), de certains MANGA choisis par un utilisateur. Le but ici est de constituer un panier. Vous devez donc proposer un algorithme bien structuré. N.B : Vous avez carte blanche : création de nouvelles fonctions, de nouveaux tableaux, de nouvelles variables, etc.