Jeune et talentueux développeur, vous intégrez les équipes d'Ubisooft (à ne pas confondre avec son concurrent), un éditeur spécialisé dans la mise à jour et le rebranding d'anciennes gloires des jeux vidéo.

Pour faire vos preuves, votre nouveau chef vous propose de reprendre un jeu qu'il a développé, lorsqu'il était stagiaire. Il sort une disquette 3,5 pouces pleine de poussière de son tiroir et vous la tend. La nostalgie voile brièvement son regard quand il vous explique que c'est une version console du Yahtzee, le célèbre jeu de dés, qu'il n'a malheureusement jamais eu le temps de terminer. Vous retrouvant, 10 minutes plus tard, en train d'essayer d'introduire la disquette dans votre Nintendo Switch, il vous explique que c'est un programme qui tourne sur la console… de Windows, l'invite de commandes.... Légèrement déçu, vous filez aux archives demander à Jean Doux un lecteur USB à brancher sur votre PC pour accéder au contenu du précieux sésame.

Vous démarrez avec le contenu de ce dossier :

- yahtzee.c, le programme principal
- yahtzee_jeu.h, la bibliothèque de fonctions gérant le déroulement du jeu
- yahtzee_affichage.h, la bibliothèque de fonctions gérant l'affichage du jeu

Vous notez qu'en compilant le programme principal les bibliothèques sont automatiquement prises en compte et intégrées à l'exécutable généré :

```
gcc yahtzee.c -o yahtzee.exe
```

Malheureusement le jeu est bugué et la compilation échoue… S'il ne se souvient pas pourquoi, votre chef se rappelle des fonctionnalités qu'il n'avait pas terminées : la mise en place d'un top 5 des scores et la sauvegarde du score en cas d'interruption du jeu.

Votre chef vous donne les objectifs suivants pour les 3 prochaines heures :

1. En parcourant le code pour comprendre son fonctionnement et sa structure, corriger l'anomalie (ou les anomalies) qui empêche la compilation,
2. Mettre en place le top 5 des scores en suivant le guide qu'il vous a laissé,
3. Si vous avez le temps, en vous inspirant de votre développement du top 5, mettre en place la sauvegarde du score.

Quoi qu'il arrive, le programme que vous livrerez doit compiler, le code doit être clair, commenté et bien indenté.

Partie 1 - Compiler et déverminer

1. En vous aidant du résultat de la compilation, corriger l'anomalie pour générer correctement l'exécutable.
2. Une fois le programme compilé, lancer l'exécutable. Après quelques tours de jeu et une chance frisant l'insolence, vous vous doutez qu'une autre anomalie s'est glissée dans le programme. Retrouvez sa cause et corrigez-la.
3. Au passage, vous observez que la fonction `int additionner_des(int *)` n'est pas définie mais juste « bouchonnée ». Terminer son développement en vous aidant des commentaires.

Partie 2 - Le Top 5

Dans cette partie, vous développerez d'abord un programme autonome, que vous transformerez ensuite en bibliothèque et intégrerez enfin au programme principal.

Voici les instructions laissées par votre chef :

Ouvrir et écrire dans un fichier

1. Créer un nouveau fichier yahtzee_top.c et ajouter les éléments de base d'un programme en C.
2. Dans la fonction principale (main), déclarer un tableau `top` de 5 entiers et l'initialiser avec 5 valeurs positives (< 390, score max du Yahtzee).

3. Ajouter la fonction suivante au programme :

```
1.  void sauvegarder_top(int * top) {
2.      FILE * fichier;
3.      int i;
4.
5.      fichier = fopen("./yahtzee.top", "w");
6.
7.
8.      fclose(fichier);
9.  }
```

Explications :

Pour manipuler les fichiers, la bibliothèque stdio.h propose un type prédéfini `FILE`, qu'on appelle flux d'accès au fichier (ou stream en anglais).

A la ligne 2, on déclare le pointeur sur FILE qui nous permettra d'accéder à ce flux.

```
FILE * fichier;
```

Pour ouvrir un fichier, on utilise la fonction `FILE * fopen(char * chemin_du_fichier, char * mode)` :

- Le premier argument est une chaîne de caractère contenant le chemin d'accès au fichier à ouvrir.
- Le second argument est une chaîne de caractère est le mode d'accès au fichier. Les principaux modes que nous utiliserons sont :

Mode	Description
r	Ouvre le fichier en lecture. Le fichier doit exister.
w	Ouvre un fichier vide en écriture. Si un fichier du même nom existe, il est écrasé.
a	Ouvre un fichier en écriture et se place à la fin du fichier. Si le fichier n'existe pas, il est créé.

La fonction renvoie un pointeur vers un FILE.

A la ligne 5, on crée un nouveau fichier yahtzee.top (ou écrase l'existant) dans le dossier de l'exécutable.

```
fichier = fopen("./yahtzee.top", "w");
```

En cas d'erreur à l'ouverture, la fonction `fopen` renvoie `NULL`.

Tout fichier ouvert doit être fermé pour le libérer.

La fonction `int fclose(FILE * flux)` permet de fermer le flux (stream) d'accès au fichier.

A la ligne 8, on ferme le flux d'accès au fichier.

```
fclose(fichier);
```

4. Compléter la fonction `sauvegarder_top` en ajoutant les instructions afin d'imprimer les données du tableau passé en argument, une valeur par ligne, en utilisant la fonction :

```
int fprintf(FILE * flux, format, arg1, arg2,... );
```

Elle fonctionne comme la fonction printf en ajoutant la variable contenant le flux d'accès au fichier en premier argument :

```
fprintf(fichier, "%d\n", var); // Imprime le contenu de la variable var dans le fichier
                               //  et passe à la ligne
```

5. Appeler la fonction `sauvegarder_top` dans votre fonction principale pour enregistrer les données du tableau déclaré à l'étape 2. Compiler, tester.

Ouvrir et lire un fichier

1. Créer la fonction `void charger_top(int *)` qui va ouvrir le fichier `"./yahtzee.top"` en lecture, lire chacune des valeurs enregistrées dans le fichier, les stocker dans le tableau passé en paramètre et fermer le flux.

   Pour lire les valeurs du fichier, on utilisera le fragment de code suivant :

```
1. char temp[5];
2. if (fichier != NULL) {
3.     while (EOF != fscanf(fichier, "%5[^\n]\n", temp)) {
4.         // La variable temp contient les 5 premiers caractères de chaque ligne
5.         // La boucle se répète autant de fois qu'il y a de lignes dans le fichier
6.     }
7. }
```

2. Modifier la fonction principale pour tester l'appel de `charger_top`. Compiler, tester.

Préparer l'intégration

Dans cette étape, nous allons simuler le chargement du top, son affichage, l'ajout d'un score dans le top, l'affichage, puis la sauvegarde du nouveau top.

1. Dans la fonction principale, initialiser le tableau `top` avec des -1.
2. Appeler la fonction `charger_top` en lui passant le tableau `top` en paramètre.
   - Si le fichier `"./yahtzee.top"` n'existe pas, il ne doit rien se passer.
3. Créer une fonction `void afficher_top(int * top, int place)` qui affiche les valeurs du tableau `top` comme suit (vide si -1) et une mention si une place est précisée :

Exemple 1 : `place` **= -1**	**Exemple 2 :** `place` **= 1**
`top`	`top`
132	212
100	132
45	131
-1	120
-1	89

`-= Top 5 des scores =-`
 	
`1. 132`
`2. 100`
`3.  45`
`4.`
`5.`
	`-= Top 5 des scores =-`
	
	`1. 212`
	`2. 132 < Vous !`
	`3. 131`
	`4. 120`
	`5.  89`

4. Dans la fonction principale, appeler la fonction `afficher_top` pour afficher le tableau `top`.
5. Créer une fonction `int mettre_a_jour_top(int * top, int score)`, qui mettra à jour le tableau `top` en insérant le `score` à la bonne place dans le tableau, décalera les valeurs suivantes (swap) et renverra l'index auquel le score à été inséré.

Exemple 1 : `score` = 132		**Exemple 2 : `score` = 132**	
`top` **avant**	`top` **après**	`top` **avant**	`top` **après**
100	132	212	212
45	100	131	132
-1	45	120	131
-1	-1	89	120
-1	-1	74	89
renvoie 0		renvoie 1	

Exemple 3 : `score` = 45	
`top` **avant**	`top` **après**
212	212
131	132
120	131
89	120
74	89
renvoie -1	

6. Dans la fonction principale, appeler `mettre_a_jour_top`, pour insérer une nouvelle valeur de votre choix.
7. Dans la fonction principale, appeler la fonction `afficher_top` pour afficher le nouveau tableau `top`.
8. Dans la fonction principale, appeler la fonction `sauvegarder_top` pour sauvegarder le nouveau tableau `top`.
9. Réaliser des tests pour éprouver votre programme.

Intégrer la nouvelle bibliothèque

Dans cette partie, nous allons transformer le programme en bibliothèque et intégrer ses fonctionnalités au programme principal.

1. Commenter la fonction principale du fichier yahtzee_top.c
2. Renommer le fichier en yahtzee_top.h
3. Inclure le fichier dans le fichier yahtzee.c en ajoutant la mention :

```
#include "yahtzee_top.h"
```

4. Récupérer les instructions de la fonction principale commentée du fichier yahtzee_top.h et les intégrer à la fonction principale du fichier yahtzee.c. Le score final est disponible dans la variable `totaux[5]`.
5. Tester.

Partie 3 - Sauvegarde du score.

Vous avez terminé en avance et souhaitez en mettre plein la vue à votre chef, vous imaginant déjà en train de développer le futur Cyberpunk 1995 (version console, compatible Windows 95, bien sûr) : en utilisant vos connaissances nouvellement acquises sur les accès aux fichiers, vous imaginez et implémentez un système qui permettra de sauvegarder à chaque tour les tableaux scores et totaux dans un fichier yahtzee.sav et proposera au joueur en cas de partie non terminée de reprendre sa partie entre deux lancements du programme.