TP C

1 Environnement de travail

Nous allons utiliser l'interface en ligne OnlineGDB, accessible via le lien suivant : `https://www.onlinegdb.com/online_c_compiler`. Elle se fonde sur le compilateur `gcc` (initialement pour « GNU C Compiler » devenu plus tard « GNU Compiler Collection » lorsqu'on l'a étendu à d'autres langages). Ce compilateur doit normalement être lancé via une ligne de commande dans un terminal. Ici, l'interface se charge de lancer la compilation et l'exécution du programme, mais nous allons modifier les options du compilateur : allez dans le menu de configuration (en haut à droite), cliquez sur « Extra Compiler Flags » et entrez l'option `-Wall`. Cette option permet d'activer un certain nombre d'avertissements du compilateur et vous aidera à écrire un code plus « propre ».

Rappel : un programme C n'est exécutable que s'il contient une fonction `main`, qui sera la fonction appelée au moment de l'exécution. Pour ce TP, nous n'utiliserons pas les arguments sur la ligne de commande donc vous pourrez écrire une fonction `main` de signature

```
int main ()
```

L'entier en retour est un code indiquant si l'exécution du programme s'est bien passée (0 le cas échéant). Vous n'observerez une sortie dans le terminal que si les instructions de la fonction `main` font usage de la fonction `printf`.

Important : tous les types et fonctions vus en cours ne sont pas directement accessibles, car ils sont répartis dans des bibliothèques. Pour y avoir accès, il faut charger les fichiers d'en-tête de ces bibliothèques. Pour ce TP, insérer les lignes suivantes au début de votre fichier sera suffisant :

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <time.h>
```

2 Échauffement

2.1 Médiane

Écrire une fonction `mediane` qui prend en entrée trois entier et qui retourne la valeur médiane de ces entiers.

2.2 Années bissextiles

Écrire une fonction `bissextile` qui prend en paramètre un entier représentant une année et qui retourne un booléen qui vaut `true` si cette année est bissextile.

On rappelle que les années bissextiles sont celles qui sont divisibles par 4, sauf celles qui sont divisibles par 100 mais pas par 400. Par exemple, 2000 est bissextile mais pas 1900.

2.3 Compte

Écrire une fonction `compte_for` qui prend un entier `n` en argument et qui affiche à l'aide d'une boucle **for** les entiers successifs de `-n` à `n` :

```
-n -(n-1) ... -1 0 1 ... n-1 n
```

Écrire une fonction `compte_while` qui fait de même à l'aide d'une boucle **while**.

3 Tableaux

3.1 Recherche d'une valeur dans un tableau

1. Écrire une fonction `cherche` prenant un tableau d'entiers et un entier en arguments et indiquant ce dernier apparaît dans le tableau. On procèdera de manière naïve en parcourant séquentiellement le tableau.
2. Écrire une fonction `tableau_aleatoire` qui prend un entier n en argument et qui renvoie un tableau de taille n rempli d'entiers aléatoires.

   Pour cela, initialisez dans la fonction `main` le générateur de nombres aléatoires à l'aide de l'instruction `srand(time(NULL));`, puis vous pourrez obtenir un entier aléatoire à l'aide de la fonction sans paramètre `rand`.
3. Implémenter un algorithme de tri sur les tableaux d'entiers.
4. Écrire une fonction `cherche_dicho` qui implémente la recherche dichotomique dans un tableau d'entiers.

3.2 Recherche d'un maximum

1. Écrire une fonction `maximum` renvoyant le maximum global d'un tableau d'entiers.

On veut maintenant trouver (efficacement) un maximum local dans un tableau, i.e. un élément qui est supérieur à l'élément précédent et à l'élément suivant, s'ils existent (attention aux bords).

2. Écrire une fonction `max_local` renvoyant l'indice d'un maximum local d'un tableau d'entiers `t`, grâce à un parcours de `t`.
3. En s'inspirant de la recherche par dichotomie, écrire une fonction `max_local_dicho` renvoyant l'indice d'un maximum local d'un tableau d'entiers.

4 Structures

Créer une structure de file contenant des entiers. On veillera à assurer que les opérations élémentaires suivantes s'exécutent en temps constant :

1. `create` : crée une file vide.
2. `add` : insère un élément dans une file.
3. `take` : extrait un élément d'une file non vide.
4. `peek` : renvoie la valeur du premier élément d'une file non vide sans l'extraire.
5. `is_empty` : teste si une file est vide.

Comment faire si on veut retourner une file ?

Pour ceux qui ont du temps : implémenter le jeu de la vie à partir d'une configuration initiale sur une grille de taille fixée (éventuellement torique).

Pour ceux qui ont vraiment du temps : simuler une grille infinie pour le jeu de la vie (à partir d'une configuration de taille finie) à l'aide d'une gestion dynamique de la mémoire.