Evaluation Langage C

Cet exercice a pour but de tester vos connaissances globales dans les concepts de variables, de structures conditionnelles, de boucles et de fonctions, ainsi que de tableaux:

1. Écrivez une fonction `int max(int a, int b)` qui retourne le maximum entre deux nombres entiers `a` et `b`.
2. Écrivez une fonction `void print_array(int arr[], int size)` qui imprime le contenu d'un tableau d'entiers de taille `size`.
3. Écrivez une fonction `int sum_array(int arr[], int size)` qui retourne la somme des éléments d'un tableau d'entiers de taille `size`.
4. Écrivez une fonction `void reverse_array(int arr[], int size)` qui inverse l'ordre des éléments d'un tableau d'entiers de taille `size`.
5. Écrivez une fonction `int find_max(int arr[], int size)` qui retourne la valeur maximale dans un tableau d'entiers de taille `size`. Utilisez la fonction `max()` écrite précédemment.
6. Écrivez une fonction `int find_min(int arr[], int size)` qui retourne la valeur minimale dans un tableau d'entiers de taille `size`. Utilisez la fonction `max()` écrite précédemment.
7. Écrivez une fonction `void sort_array(int arr[], int size)` qui trie les éléments d'un tableau d'entiers de taille `size` dans l'ordre croissant. Utilisez la fonction `max()` et la boucle `for` pour implémenter l'algorithme de tri par sélection.

NB

Tous les prototypes de vos fonctions devront être dans le fichier `main.h` et les fonctions seront testées avec le script suivant:

```
#include <stdio.h>
#include "main.h"

int main(void) {
  int arr[] = {5, 2, 7, 1, 8, 3};
  int size = sizeof(arr) / sizeof(int);

  printf("Tableau original : ");
  print_array(arr, size);

  printf("Somme des éléments du tableau : %d\n", sum_array(arr, size));

  reverse_array(arr, size);
  printf("Tableau inversé : ");
  print_array(arr, size);

  printf("Valeur maximale du tableau : %d\n", find_max(arr, size));
  printf("Valeur minimale du tableau : %d\n", find_min(arr, size));

  sort_array(arr, size);
  printf("Tableau trié : ");
  print_array(arr, size);

  return 0;
}
```

Le compilateur utilisé sera GCC. Il y aura quatre niveaux de compilation.

1. `gcc *.c`
2. `gcc -Wall -Werror *.c`
3. `gcc -Wall -Werror -pedantic -Wextra *.c`
4. `gcc -Wall -Werror -pedantic -Wextra -std=gnu89 *.c`

Pour les trois premiers niveaux, chaque niveau passé a donne plus de points que le précédent. Le quatrième est un niveau bonus. Vous pouvez faire des recherches sur les différentes options du compilateur gcc `-Wall` `-pedantic` `-Werror` `-Wextra` `-std=gnu89`. De plus, **La compilation à chaque niveau devra se faire sans aucun avertissement ni erreur**. Après compilation, l'exécutable sera testé comme suit: `./a.out` et la sortie devra être telle que ci-contre:

```
Tableau original : 5 2 7 1 8 3
Somme des éléments du tableau : 26
Tableau inversé : 3 8 1 7 2 5
Valeur maximale du tableau : 8
Valeur minimale du tableau : 1
Tableau trié : 1 2 3 5 7 8
```

Il faut noter que ces tests ne sont pas les seuls qui seront effectués sur votre code. Pensez donc à prévoir des cas non pris en charge ici.

Ressources Utiles:

- GCC Warning Options
- GCC Options Controlling C Dialect
- Everything you need to know to start with C