Runtrack C : Jour 3

Job 01

Vous devez implémenter la fonction **my_atoi** (dans un fichier **my_atoi.c**) qui va prendre en paramètre une chaîne de caractères, et qui doit la convertir en int.

```
int my_atoi(char *str);
```

Fonctions autorisées : **aucune**.

Job 02

Vous devez implémenter la fonction **my_itoa** (dans un fichier **my_itoa.c**) qui prend en paramètre un int, et qui doit le convertir en chaîne de caractères.

```
char *my_itoa(int n);
```

Fonctions autorisées : **malloc**.

Job 03

Vous devez implémenter la fonction **power** (dans un fichier **power.c**) qui doit prendre en paramètre deux int, et élever le premier à la puissance du deuxième.

```
int power(int n, int power);
```

Fonctions autorisées : **aucune**.

Job 04

Vous devez implémenter la fonction **my_sqrt** (dans un fichier **my_sqrt.c**) qui prend en paramètre un int, et retourne la racine carrée de cet int, ou 0 si la racine carrée n'est pas un entier.

```
int sqrt(int n);
```

Fonctions autorisées : **aucune**.

Job 05

Vous devez créer un programme qui prend en argument un entier, et écrit dans la sortie standard la factorielle de cet entier.

Si le programme ne reçoit pas d'arguments, il ne doit rien faire.

```
./job5.exe 3
6
```

Fonctions autorisées : **write**.

Job 06

Vous devez créer un programme qui prend en argument un entier n, et écrit dans la sortie standard le nième nombre de la suite de fibonacci.

Si le programme ne reçoit pas d'arguments, il ne doit rien faire.

Fonctions autorisées : **write**.

Job 07

Vous devez implémenter la fonction **itoa_hex** (dans un fichier **itoa_hex.c**) qui prend en paramètre un int et le convertit en chaîne de caractères, en base hexadecimale (avec des caractères alphabétiques en majuscules).

Fonctions autorisées : **malloc**.

Job 08

Vous devez implémenter la fonction **sudoku_solver** (dans un fichier **sudoku_solver.c**) qui prend en paramètre une grille de sudoku sous forme

de tableau d'int à deux dimensions, tente de résoudre la grille, renvoie 1 si elle a une solution et 0 si elle n'en a pas.

```
int sudoku_solver(int grid[9][9]);
```

Fonctions autorisées : **aucune**.

Rendu

Le projet devra être rendu sur votre github, dans un repository nommé **runtrack_c**. Le repo doit contenir un dossier pour chaque jour de la runtrack, nommés "Jour01", "Jour02", "Jour03", "Jour04", "Jour05".
Chacun de ces dossiers devra contenir les jobs dans le dossier respectif : "Job01", "Job02", etc.
Les prototypes des fonctions doivent toujours être exactement les mêmes que dans les énoncés.

Compétences visées

- C

Base de connaissances

- https://www.w3schools.com/c/c_getstarted.php
- https://openclassrooms.com/fr/courses/19980-apprenez-a-programmer-en-c
- https://www.my-mooc.com/fr/mooc/c-programming-getting-started