****

**Runtrack C : Jour 1**

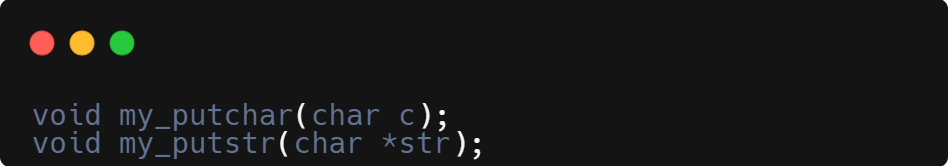
*Parce que des fois les String, ça n’existe pas.*

# Job 01

Vous devez implémenter deux fonctions, my\_putchar (dans un fichier my\_putchar.c) et my\_putstr (dans un fichier my\_putstr.c)

La fonction my\_putchar prend un caractère en paramètre, et l’écrit sur la sortie standard.

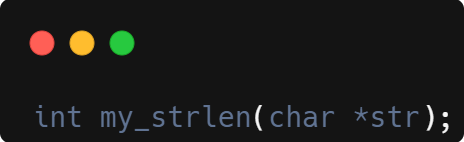
La fonction my\_putstr prend une chaîne de caractères en paramètre, et l’écrit sur la sortie standard, en utilisant la fonction my\_putchar.



Fonctions autorisées : write.

# Job 02

Vous devez implémenter la fonction my\_strlen (dans un fichier my\_strlen.c), qui prend en paramètre une chaîne de caractères, et qui renvoie sa taille.



fonctions autorisées : aucune.

# Job 03

Vous devez implémenter la fonction char\_is\_digit (dans un fichier char\_is\_digit.c) qui prend en paramètre un caractère, et qui renvoie 1 si le caractère est un chiffre, 0 si ça n’en est pas un.  
Vous devez aussi implémenter une fonction str\_is\_digit (dans un fichier str\_is\_digit.c) qui prend en paramètre une chaîne de caractères, et qui renvoie 1 si la chaîne ne contient que des chiffres, 0 si ce n’est pas le cas.



Fonctions autorisées : aucune.

# Job 04

Vous devez implémenter la fonction my\_strcpy (dans un fichier my\_strcpy.c), qui prend en paramètres une chaîne de caractères de destination et une de source, et qui copie la source dans la destination.



Fonctions autorisées : aucune.

# 

# Job 05

Vous devez implémenter la fonction my\_strdup (dans un fichier my\_strdup.c) qui prend en paramètre une chaîne de caractères, alloue la mémoire nécessaire pour une copie de la chaîne entrante, copie celle-ci dans l’espace mémoire fraîchement alloué.



Fonctions autorisées : malloc.

# Job 06

Vous devez implémenter la fonction my\_strcmp (dans un fichier my\_strcmp.c) qui prend en paramètre deux chaînes de caractères, les compare, et renvoie 0 si elles sont identiques, et une valeur négative si la première chaîne est inférieure à la deuxième, ou positive si elle est supérieure à la deuxième.



Fonctions autorisées : aucune.

Note : Une chaîne est inférieure à une autre si le premier caractère différent dans les deux chaînes est inférieur (dans l’ordre de la table ASCII) dans la première chaîne par rapport à celui de la deuxième chaîne.

Par exemple, la chaîne “abc” est inférieure à la chaîne “abd” parce que les deux premiers caractères sont identiques, mais “c” est inférieur à “d” dans la table ASCII.

# Job 07

Vous devez implémenter la fonction trim (dans le fichier trim.c) qui prend en paramètre une chaîne de caractères, alloue la mémoire nécessaire pour une copie de la chaîne entrante, en retirant les espaces en début et fin de chaîne, puis copie la source dans la nouvelle chaîne.



Fonctions autorisées : malloc.

Par exemple, la chaîne “ hello world “ deviendra “hello world”.

Les espaces comprennent aussi les tabulations et les retours à la ligne.

# Job 08

Vous devez implémenter la fonction split (dans le fichier split.c) qui prend en paramètre une chaîne de caractères, et renvoie un tableau de chaînes de caractères, résultat de la séparation de la chaîne entrante séparée par mots, et se termine par un pointeur NULL.

Fonctions autorisées : malloc.

Par exemple, la chaîne “ hello world have fun” renvoie un tableau contenant “hello”, “world”, “have”, “fun”, et NULL.

Les espaces comprennent aussi les tabulations et les retours à la ligne.

# 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.

Les fichiers seront compilés avec les flags -Wall -Wextra -Werror.

# Compétences visées

* C

# Base de connaissances

* man