

Exercice 501

Enoncé

- Qu'est-ce qu'un booléen ? Quand obtient-on une réponse booléenne ?
- En C, existe-t-il un type booléen. Si non, comment simule-t-on un booléen ?
- En C, quelle est la valeur booléenne de la lettre 'k', du caractère '0' (zéro) et du chiffre 0 ?

Corrigé

Le code se trouve dans le fichier [main.c](#).

Réponses

- Un booléen est un type de variable qui a deux états possibles : `True` ou `False`. On obtient une réponse booléenne, par exemple lorsqu'on compare deux valeurs avec un opérateur de comparaison. Ex: `5 > 3` renvoie `True` car 5 est bien supérieur à 3.
- En C il existe bien un type booléen, mais il faut inclure la bibliothèque `stdbool.h` pour pouvoir l'utiliser. On peut alors déclarer une variable booléenne avec `bool` et lui affecter `true` ou `false`. Ex: `bool a = true;`. L'avantage de ce type de variable est qu'il ne prend qu'un bit en mémoire (2^1 donne bien 2 valeurs possibles), contrairement à un `int` qui prend 4 octets (ou 32bits). On peut donc économiser de la mémoire en utilisant des booléens. Pour simuler un booléen on peut prendre un `int` et ne jamais lui affecter que la valeur `1` ou `0` pour le traiter comme un bool. On peut aussi utiliser un `char` et lui affecter `'T'` ou `'F'` par exemple. Ceci est plus économe en mémoire car un `char` n'utilise qu'un seul octet en mémoire (cf. Exo 405). NOTONS QUE LE BOOL EXISTE DEJA DANS STDBOOL.H
- la lettre 'k' et le caractère '0' ont tous une valeur booléenne de `True`. Le chiffre 0 en revanche, vaut bien `False`. (Preuve: [Code Exo 501](#))