Week-end Challenge
01

Subject:

For this challenge, you will be asked to create your own morse code interpreter.

Rules:

- Only 1 submission per student
- Your code has to be written in C and compile with the following flags: `-Wall -Wextra -Werror`
- Try to respect the norm as much as you can to make it readable, but also feel free to take some liberties if you want to: longer functions (more than 25 lines), the use of `switch` and `for`, etc
- To compile your program, please provide a `Makefile` or, if you can't be bothered, a script called `build.sh`.
- You do not have to handle error cases (invalid morse string), but doing it will be considered a bonus.

Mandatory part:

program name	morsec
Turn-in files	*.c / *.h / Makefile or build.sh
Arguments	the morse string
Authorized libraries	anything from the **libc**
Description	Your program should receive morse code as a string made of "-./ ",translate it to text and displaying it in **stdout.**

Your interpreter will be able to read letters according to this dictionary. Morse letters will be separated by spaces and words will be separated by a slash. Example:

```
> ./morsec "-- -.-- / -. .- -- . / .. ... / .--- . ..-. ..-."
my name is jeff
>
```

Ressources:

- WEC GitHub repository
- Morse code dictionary

Bonus:

If your program works perfectly, you can add more features to increase your chances to win. In this case, the bonus will be able to add a reverse option to your program.
Example:

```
> ./morsec -r "vibe check"
...- .. -... . / -.-. .... . -.-. -.-
>
```

Turn-in:

To submit your project, compress the folder of the source code into a zip archive and send it by email to tutors@s19.be with the following subject: `wec01 - [login]`. The deadline is Sunday at 10:19pm.

Results:

The results will be announced the following Monday's AMA, and **3500** points will be shared between the winners.