KEEP CALM & USE FUNCTIONS

SCRIPT DE COMMANDE

```
const face = (d:Direction) : Command => ({kind: "Face", direction: d})
const start = () : Command => ({kind: "Start"})
const stop_ = () : Command => ({kind: "Stop"})
const chain = (first: Command, second: Command) : Command => ({kind: "Chain", first, second})

const pipe = (first: Command, ...rest: Array<Command>) : Command =>
    rest.length > 0
    ? chain(first, pipe(rest[0], ...rest.slice(1)))
    : first
```

MIAGE M2 - QUALITÉ DU SI - THOMAS HAESSLÉ

```
const moveWestNorth = pipe(
    face("West"),
    start(),
    stop_(),
    face("North"),
    start(),
    stop_()
)
```

SCRIPT DE COMMANDE

KEEP CALM & USE FUNCTIONS 

```
const face = (d:Direction) : Command => ({kind: "Face", direction: d})
const start = () : Command => ({kind: "Start"})
const stop_ = () : Command => ({kind: "Stop"})
const chain = (first: Command, second: Command) : Command => ({kind: "Chain", first, second})

const pipe = (first: Command, ...rest: Array<Command>) : Command =>
    rest.length > 0
    ? chain(first, pipe(rest[0], ...rest.slice(1)))
    : first
```

```
const moveWestNorth = pipe(
    face("West"),
    start(),
    stop_(),
    face("North"),
    start(),
    stop_()
)
```

TAKE AWAY

NOUS AVONS VU

Le type Command est récursif : s'exprime en terme de {kind: "Chain", first Command, second Command}

Un système qui a des types « ET », des types « OU » et des types récursifs s'appelle un système de données algébriques (ADT)

Un ADT permet de représenter les valeurs autorisées d'un programme et leurs transitions