PP => D+N=AWESOME

Ladies and Gentlemen, Start your cogitation engines...

```
if (wkstnNum%2==1) {
    thinker.login();
    thinker.open(codeEditor);
    thinker.open(terminal);
} else {
    thinker.fetch(KtS);
    thinker.fetch(KtS);
    thinker.craft(teamFlag);
}
```

Analog feedback on rally racing?

a typical rallycross scene...

We aim to avoid this

...and instead aim for this:

pair programming: driver + navigator = awesome!

pair programming: driver + navigator = awesome!

- only 1 Thinker ("driver") types
- other Thinker ("co-driver") observes/guides/checks/co-creates
- both thinkers *think*
- both thinkers *get smarter faster*
- swap roles at specified intervals

~~Ladies and Gentlemen~~, *// <<< it's a spectrum >>>*

Thinkers,

Start your cogitation engines…

- *driver: logged in, ready to code*
- *co-driver: KtS on hand, ready to "navigate"*

rules of the road:

- ***raise team flag*** to indicate
  - distress
  - something to show off
- ***listen*** for signal to
  - swap seats
  - pause for announcements

...aaaand they're off! (it’s NOT A RACE)

1. **bar_graphify(nums)** takes a list of non-negative integers and prints a horizontal bar for each index, commensurate with the value at said index. *Exempli gratia,*

```
>>> x = [0,1,2,3]

>>> bar_graphify(x)

0:

1: =

2: ==

3: ===
```

```
>>> x = [1,0,3,2]

>>> bar_graphify(x)

0: =

1:

2: ===

3: ==
```

GOooo!!!! (it’s NOT A RACE)

2. **vertbargraphify(nums)** takes a list of non-negative integers and prints a set of vertical bars visualizing the magnitude of the value at each index. *Exempli gratia,*

```
>>> x = [0,1,2,3]
>>> vertbargraphify(x)
      *
    * *
  * * *
0 1 2 3
```

```
>>> x = [1,0,3,2]
>>> vertbargraphify(x)
    *
    * *
*   * *
0 1 2 3
```

pedagogical notes

=======================

pair programming: deploy wisely

Good for:

- Accelerating learnination of
  - **new tools**
  - **language syntax**
  - **platforms**
  - **new techniques**
- Encouraging collaboration
- Realizing fallibility
- Building sense of community

Less ideal for:

- Big conceptual challenges
- Multi-faceted, large-scope projects

...but with modified structure,
can still work well.

pp PROTIPS

- play the referee/support van role...
  - enforce 1-driver rule
  - respond to distress signals
  - watch for less-visible cues, respond appropriately
  - ***encourage***
- visible timer
- clear signal for transition
- "Hints at the Half"
- **kiss**: odds drive first
- **physically** switch seats
- **notebook <-> map**

- violation/hatchback/teamflag/siren/etc… milk the rally metaphor...
- **“Not a race. No really tho.”**

resources + miscellany

background: https://en.wikipedia.org/wiki/Pair_programming

convince your uber-nerds: https://www.newyorker.com/magazine/2018/12/10/the-friendship-that-made-google-huge

amuse: Samir, you're breaking the car *...w/ follow-ups 1 and 2*

inspire: Rally driving done well

what is AWESOME? see Fake Grimlock page & twitter