

Debugging Remote Systems

Saving Time and Keystrokes

Amos King <amos@binarynoggin.com> - Binary Noggin - Omaha Elixir Feb. 2023

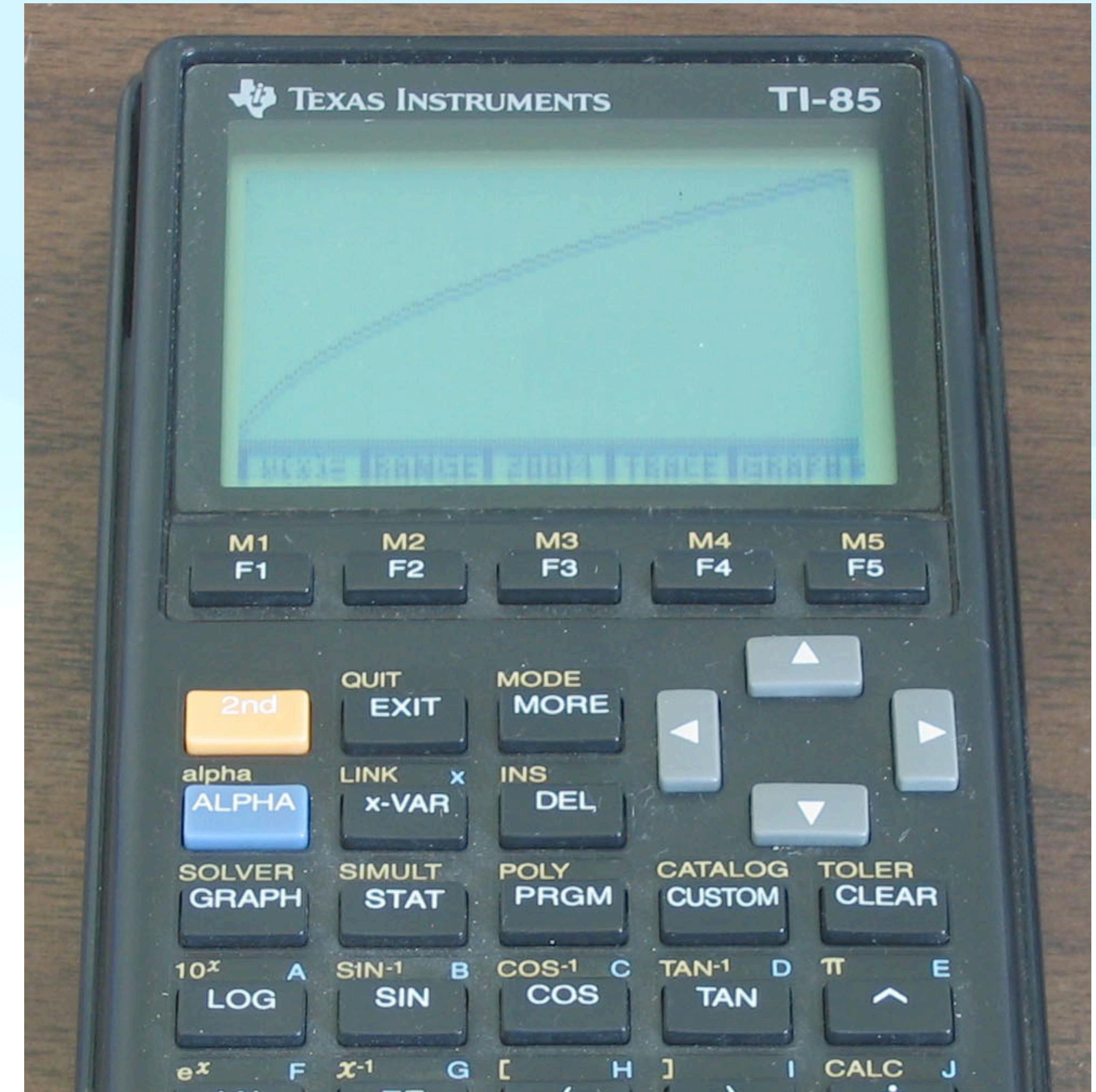
Who Am I?

Amos King

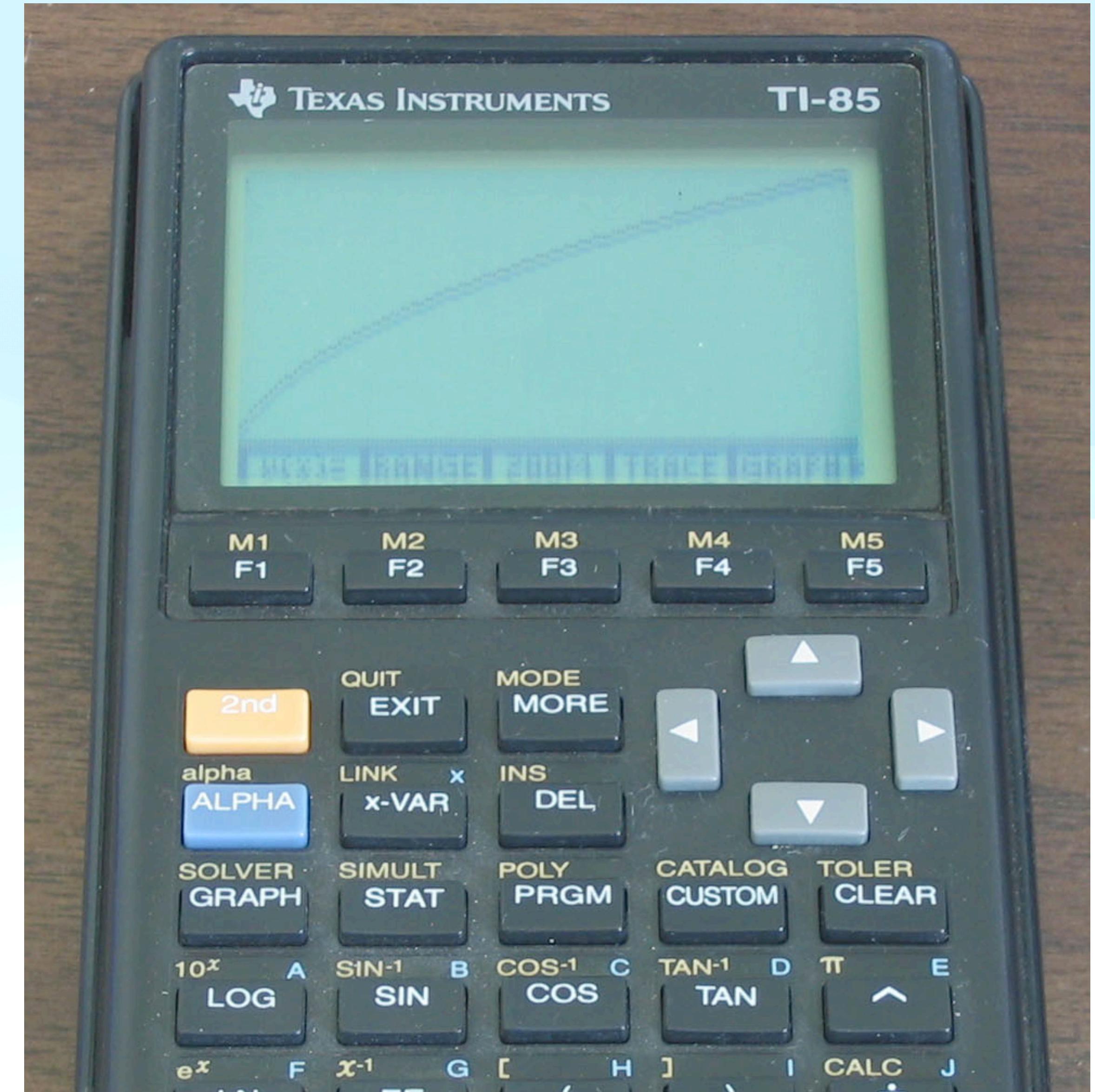
- Founder of Binary Noggin
 - <https://binarynoggin.com>
- Host of Elixir Outlaws
 - <Https://elixiroutlaws.com>
- Father of Five
- amos@binarynoggin.com

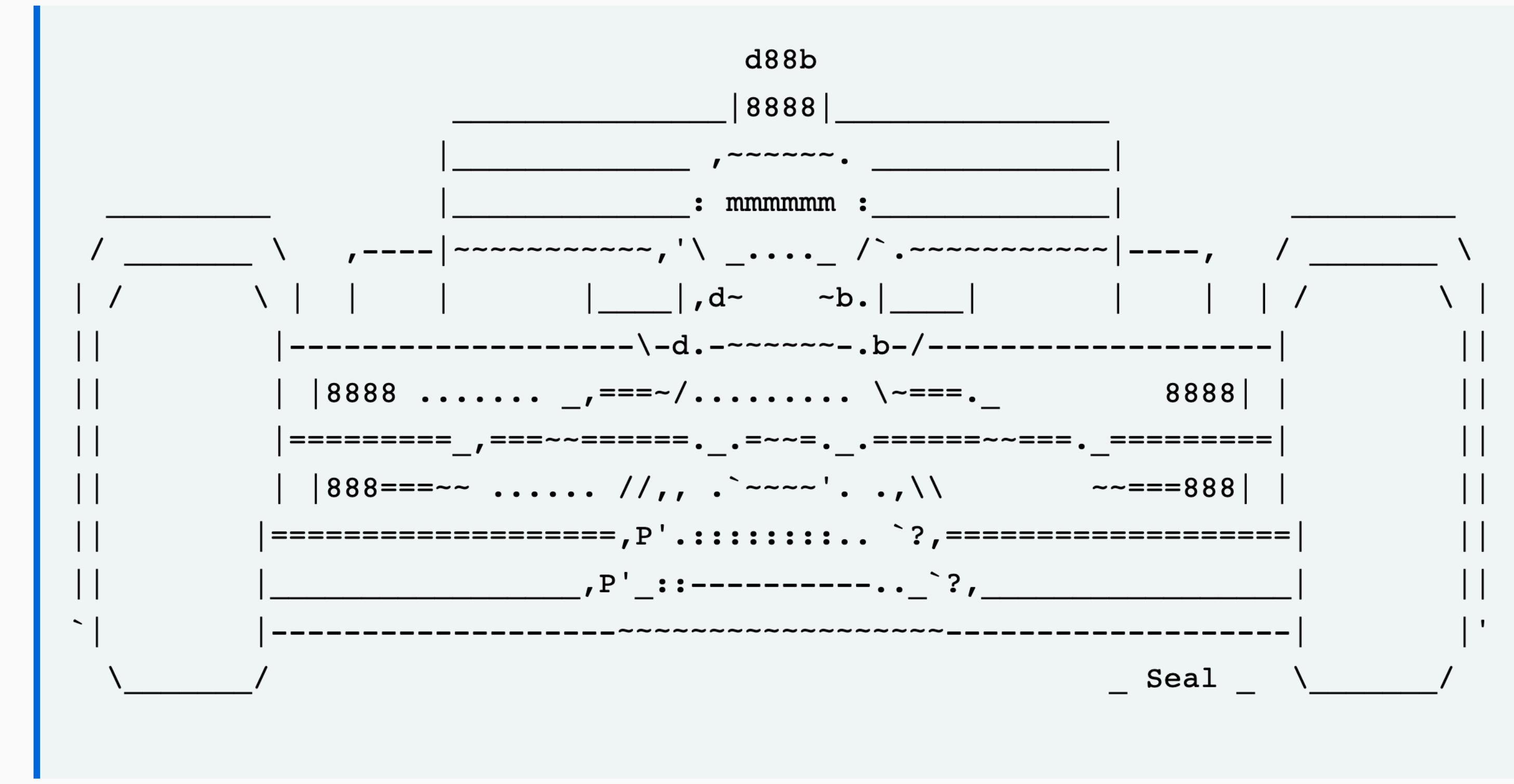


Do you remember this?

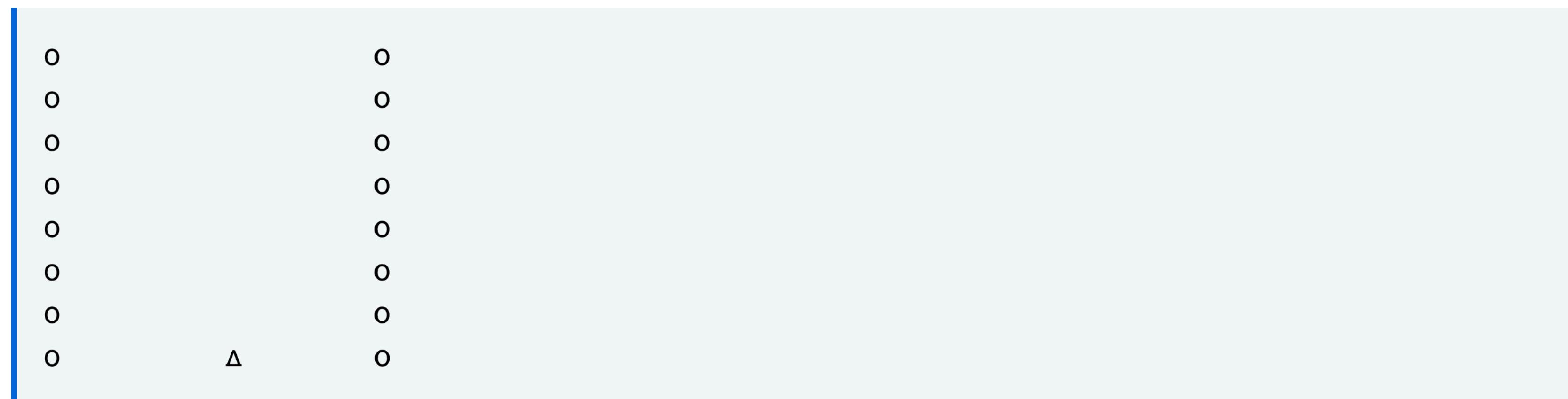


If you do,
congratulations
you are old.





Score: 0 Speed: 0 Obstacles: true Warp: false



Demo

“You’re hired. Welcome to the team!”

Pointy Haired Boss

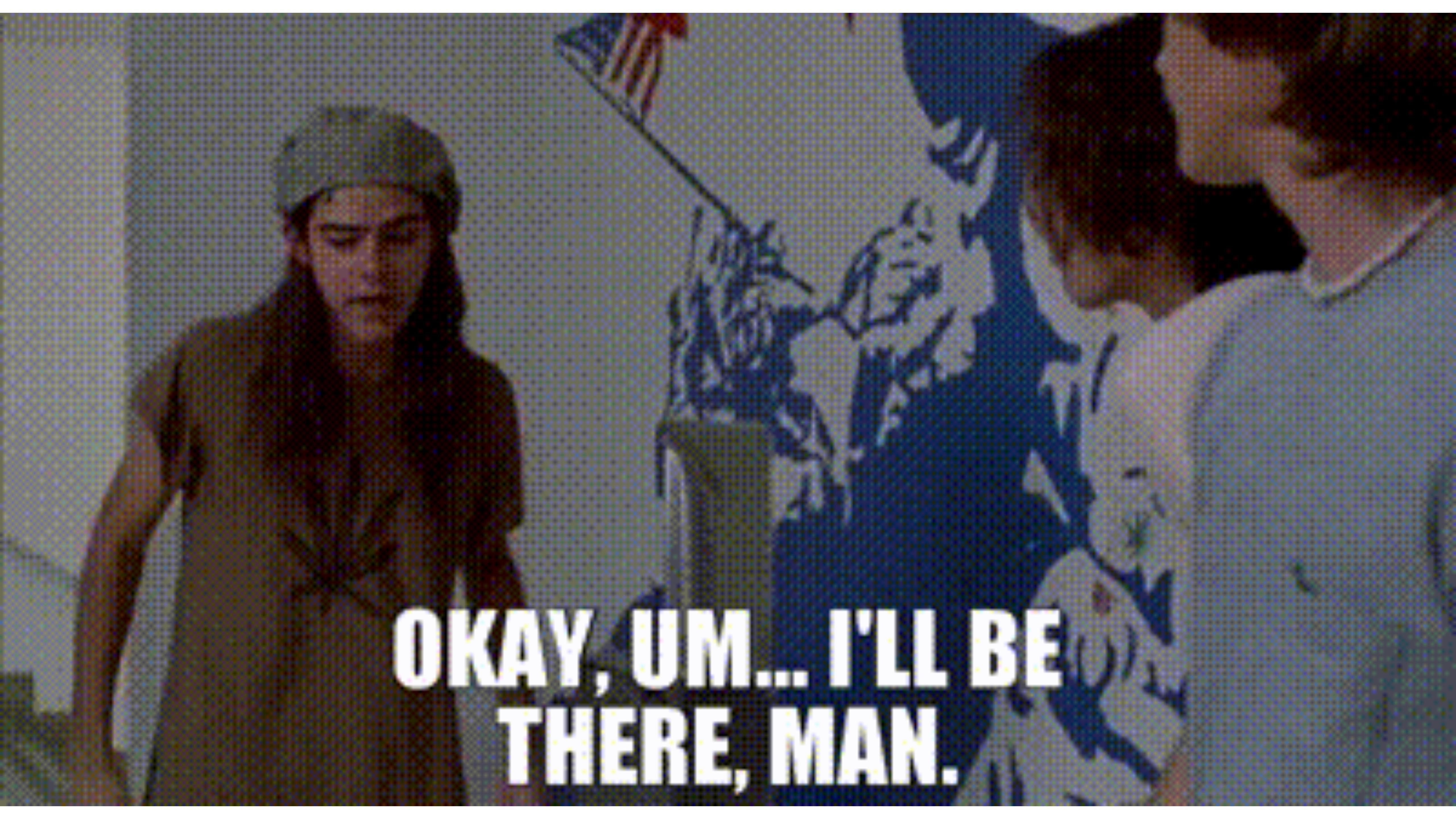
A close-up photograph of a man with short brown hair, smiling broadly with his eyes closed. He is wearing a light blue t-shirt. In his right hand, he holds a dark-colored book with a yellow label on the spine that reads "The". In his left hand, he holds another book with a red cover. The background is dark and out of focus.

Alright, Alright, Alright

[https://github.com/
BinaryNoggin/text_racer](https://github.com/BinaryNoggin/text_racer)

**“Sorry about this. I know it is your first day,
but the game is broken. Why don’t you
see what you can do?”**

Pointy Haired Boss



OKAY, UM... I'LL BE
THERE, MAN.

Debugging

- Venturing into the unknown
- It takes time and money
- Built-in Tools

Built-in Tools

But it was free

- IO.inspect/3
- Kernel.debug/2
- IEx.pry/0
- IEx Helpers - i, t, exports, h, paste

Demo



30-50%

Time Spent Debugging

What is your debugging budget?



50-75%

\$\$ Total Budget Spent on Debugging

A medium shot of a woman with long dark hair, wearing a dark jacket over a patterned top, looking towards the right. A man with short brown hair, wearing a purple shirt, is partially visible behind her, looking back at her. The background is a dark, textured wall.

like how?

:recon_trace

Production Safe Debugging

https://hexdocs.pm/recon/recon_trace.html



Installing Recon

mix deps.install

```
{:recon, "~> 2.5.3"}
```

Secure Connections

- Cookies
- REMSH
- EPMD
 - Jump Box - SSH
 - SSH to Node
- Erlang



You cool, man?

Running Server Locally

```
iex --name "production@127.0.0.1"  
--cookie secret  
-S mix phx.server
```

Connecting to a local Server

```
iex --name "amos@local"  
--cookie secret  
-remsh "production@127.0.0.1"
```

ReconTrace

Calls/3

```
:recon_trace.calls(tspec, max, options)
```

ReconTrace.tspec

{mod, fun, args}

```
@spec tspec :: {mod, f, args}
```

```
@spec mod :: :_ | module
```

```
@spec f :: :_ | atom
```

```
@spec args :: :_ | 0..255 | matchspec | shellfun
```

ReconTrace

Calls/3 - wild cards

```
iex> :recon_trace.calls({:_, :_, :_}, 10)
```

ReconTrace

Calls/3 - wild cards

```
iex> :recon_trace.calls({:_, :_, :_}, 10)  
** (ErlangError) Erlang error:  
{:dangerous_combo, {:_, :_, :_}}
```

ReconTrace

Calls/3 - multiple modules

```
:recon_trace.calls(  
[ {TextRacerWeb.GameLive, :_, :_},  
{TextRacer.Game, :_, :_},  
{TextRacer.Row, :_, :_} ],  
100, args: :arity)
```

ReconTrace

Calls/3 - specific function

```
:recon_trace.calls(  
{TextRacerWeb.GameLive, :handle_info, _:},  
10)
```

ReconTrace

Calls/3 - filtering by arguments - shellfun

```
tick_event_filter =
fn([_, %{assigns: %{game: %{speed: speed}}}]) when speed > 0 →
    :ok
end

:recon_trace.calls(
{TextRacerWeb.GameLive, :handle_info, tick_event_filter},
10)
```

ReconTrace

Calls/3 - formatter

```
:recon_trace.calls(  
  {TextRacerWeb.GameLive, :handle_info, tick_event_filter},  
  10,  
formatter: fn(args) → IO.inspect(args) end)
```

ReconTrace

Calls/3 - filtering by function name

```
:recon_trace.calls(  
  {TextRacer.Row, :update, :_}, 10)
```

ReconTrace

Calls/3 - filtering by arguments - match specs

```
shrink_matcher = [
  {
    [:_, :shrink],
    [],
    []
  }
]
:recon_trace.calls({TextRacer.Row, :update, shrink_matcher}, 10)
```

ReconTrace

Calls/3 - return values

```
shrink_matcher = [
  {
    [:_, :shrink],
    [],
    [{:return_trace}]
  }
]
:recon_trace.calls({TextRacer.Row, :update, shrink_matcher}, 10)
```

ReconTrace

Calls/3 - narrowing down

```
shrink_matcher = [
  {
    %{{left: :"$1", min_width: :"$3", right: :"$2"}, :shrink},
    [ {":=<", {:-, :"$2", :"$1"}, :"$3"} ],
    [:return_trace]
  }
]

:recon_trace.calls({TextRacer.Row, :update, shrink_matcher}, 10)
```

Ex2ms

Matchspecs are tough

```
import Ex2ms
shrink_matcher =
fun do
  { %{left: l, right: r, min_width: m}, :shrink }
  when r - l ≤ m →
    { :return_trace }
end
```

Options

```
[  
  pid: pid_spec,  
  args: :arity | :args,  
  scope: :global | :local  
  formatter: (term → iodata)  
]
```

Remote Connections

- Use `mix release`
- `app_name remote`



Demo

Recon

Diagnose Problems on Production Nodes

- `:recon` - high-level functionality
- `:recon_alloc` - discover problems with Erlang memory allocators
- `:recon_lib` - digging in and getting fancy
- `:recon_map` - formatting maps for output by other recon modules
- `:recon_rec` - formatting records for output by other recon modules
- `:recon_trace` - production safe debugging

Resources

Recon - <https://hexdocs.pm/recon/overview.html>

The Debugging Mindset - Devon H. O'Dell - ACM Queue Jan-Feb 2017 <https://queue.acm.org/detail.cfm?id=3068754>

Erlang In Anger - Fred Hebert - <https://www.erlang-in-anger.com/>

Matchspec - https://www.erlang.org/doc/apps/erts/match_spec.html

Ex2ms - <https://hexdocs.pm/ex2ms/readme.html>

Erlang's remsh is dangerous - <https://broot.ca/erlang-remsh-is-dangerous.html>