WHAT WOULD SOLIDITY LOOK LIKE IF IT WAS BUILT TODAY?

CODE EXAMPLE

```solidity
interface Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable;
}

contract Foo is Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable override {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```solidity
interface Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable;
}

contract Foo is Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable override {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
abi Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable;
}

impl Mintoooor for Foo {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}

impl Foo {
}
```

```
abi Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable;
}

impl Mintoooor for Foo {
    function mint(uint8 v, bytes32 r, bytes32 s) external payable {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}

impl Foo {
}
```

```
abi Mintoooor {
    #[payability(nonpayable)]
    function mint(uint8 v, bytes32 r, bytes32 s) external;
}

impl Mintoooor for Foo {
    #[payability(nonpayable)]
    function mint(uint8 v, bytes32 r, bytes32 s) external {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
abi Mintoooor {
    #[payability(nonpayable)]
    function mint(uint8 v, bytes32 r, bytes32 s) external;
}

impl Mintoooor for Foo {
    #[payability(nonpayable)]
    function mint(uint8 v, bytes32 r, bytes32 s) external {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
abi Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external;
}

impl Mintoooor for Foo {
    function mint(uint8 v, bytes32 r, bytes32 s) external {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```sway
abi Mintoooor {
    function mint(uint8 v, bytes32 r, bytes32 s) external;
}

impl Mintoooor for Foo {
    function mint(uint8 v, bytes32 r, bytes32 s) external {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        address a = ecrecover(MAGIC, v, r, s);
        // mint to `a`
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        Result r = ecrecover(MAGIC, v, r, s);
        address a = r.unwrap();
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        Result r = ecrecover(MAGIC, v, r, s);
        address a = r.unwrap();
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        address a = ecrecover(MAGIC, v, r, s).unwrap();
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(uint8 v, bytes32 r, bytes32 s) {
        address a = ecrecover(MAGIC, v, r, s).unwrap();
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(v: uint8, r: bytes32, s: bytes32);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(v: uint8, r: bytes32, s: bytes32) {
        let a: address = ecrecover(MAGIC, v, r, s).unwrap();
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(v: uint8, r: bytes32, s: bytes32);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(v: uint8, r: bytes32, s: bytes32) {
        let a: address = ecrecover(MAGIC, v, r, s).unwrap();
    }
}
```

```
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    function mint(v: uint8, r: bytes32, s: bytes32);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    function mint(v: uint8, r: bytes32, s: bytes32) {
        let a = ecrecover(MAGIC, v, r, s).unwrap();
    }
}
```

```sql
use std::crypto::ecrecover;

abi Mintoooor {
    #[visibility(external)]
    fn mint(v: u8, r: b256, s: b256);
}

impl Mintoooor for Foo {
    #[visibility(external)]
    fn mint(v: u8, r: b256, s: b256) {
        let a = ecrecover(MAGIC, v, r, s).unwrap();
    }
}
```

FIN