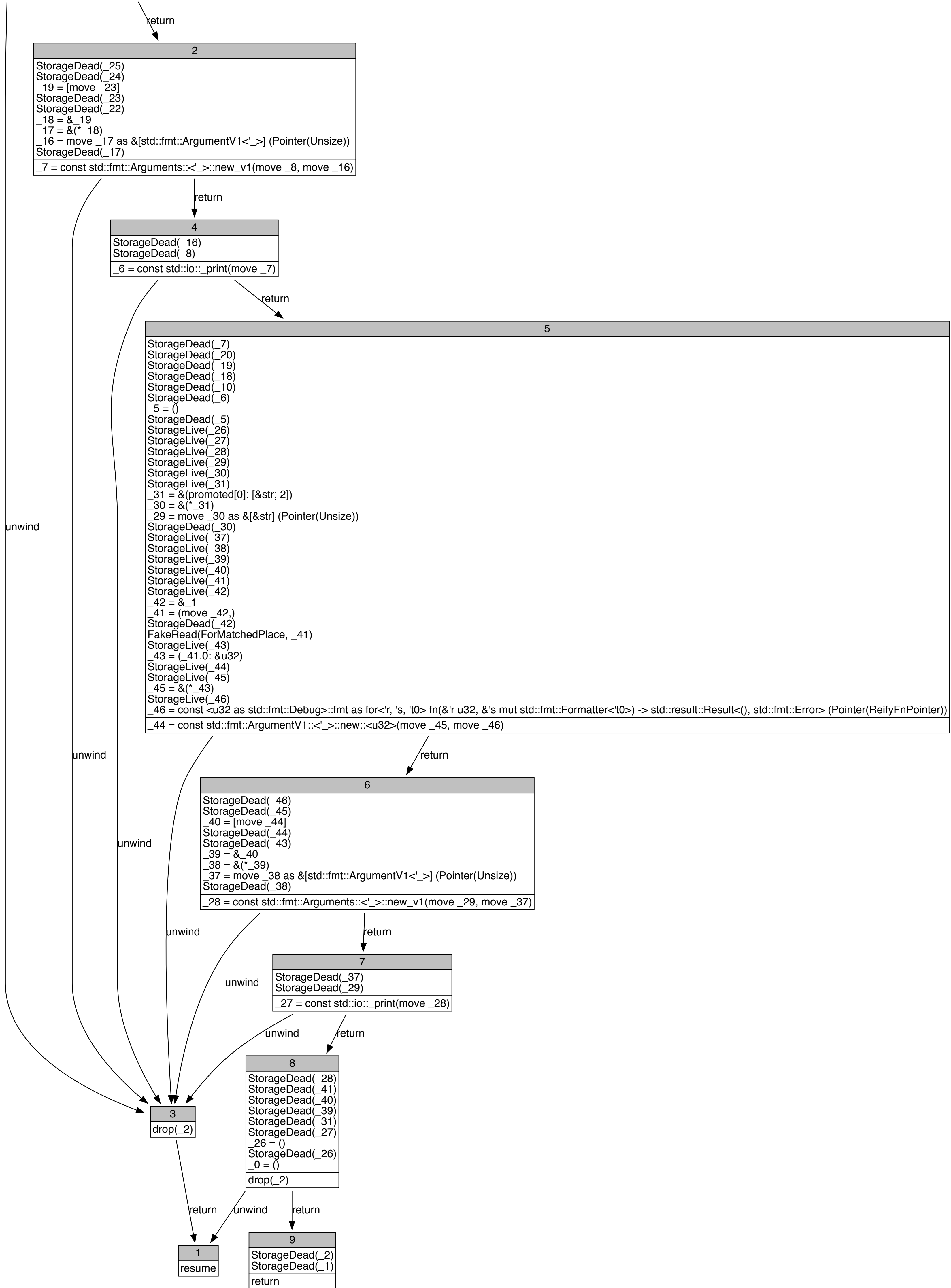


```

0
StorageLive(_1)
_1 = const 13u32
FakeRead(ForLet, _1)
StorageLive(_2)
StorageLive(_3)
StorageLive(_4)
_4 = &_1
_3 = &(*_4)
_2 = Foo::<_> { data: move _3 }
StorageDead(_3)
FakeRead(ForLet, _2)
StorageDead(_4)
StorageLive(_5)
StorageLive(_6)
StorageLive(_7)
StorageLive(_8)
StorageLive(_9)
StorageLive(_10)
_10 = &(promoted1): [&str; 2]
_9 = &(*_10)
_8 = move _9 as &[&str] (Pointer(Unsize))
StorageDead(_9)
StorageLive(_16)
StorageLive(_17)
StorageLive(_18)
StorageLive(_19)
StorageLive(_20)
StorageLive(_21)
_21 = &_2
_20 = (move _21,)
StorageDead(_21)
FakeRead(ForMatchedPlace, _20)
StorageLive(_22)
_22 = (_20.0: &Foo<_>)
StorageLive(_23)
StorageLive(_24)
_24 = &(*_22)
StorageLive(_25)
_25 = const <Foo<_> as std::fmt::Debug::fmt as for<r, 's, 't0> fn(&r Foo<_>, &s mut std::fmt::Formatter<t0>) -> std::result::Result<(), std::fmt::Error> (Pointer(ReifyFnPointer))
_23 = const std::fmt::ArgumentV1::<_>::new::<Foo<_>>(move _24, move _25)

```



```

fn main() -> ()
let _1: u32; // x
let _2: Foo<_>; // y
let mut _3: &u32;
let mut _4: &u32;
let _5: ();
let _6: ();
let mut _7: std::fmt::Arguments<_>;
let mut _8: &[&str];
let mut _9: &[&str; 2];
let mut _10: &[&str; 2];
let _11: [&str; 2];
let mut _12: &str;
let mut _13: &str;
let mut _14: &str;
let mut _15: &str;
let mut _16: [std::fmt::ArgumentV1<_>];
let mut _17: [std::fmt::ArgumentV1<_>; 1];
let mut _18: [std::fmt::ArgumentV1<_>; 1];
let _19: [std::fmt::ArgumentV1<_>; 1];
let mut _20: (&Foo<_>);
let mut _21: &Foo<_>;
let mut _22: &Foo<_>; // arg0
let mut _23: std::fmt::ArgumentV1<_>;
let mut _24: &Foo<_>;
let mut _25: for<r, 's, 't0> fn(&r Foo<_>, &s mut std::fmt::Formatter<t0>) -> std::result::Result<(), std::fmt::Error>;
let _26: ();
let _27: ();
let mut _28: std::fmt::Arguments<_>;
let mut _29: [&str];
let mut _30: [&str; 2];
let mut _31: [&str; 2];
let _32: [&str; 2];
let mut _33: &str;
let mut _34: &str;
let mut _35: &str;
let mut _36: &str;
let mut _37: [std::fmt::ArgumentV1<_>];
let mut _38: [std::fmt::ArgumentV1<_>; 1];
let mut _39: [std::fmt::ArgumentV1<_>; 1];
let mut _40: [std::fmt::ArgumentV1<_>; 1];
let mut _41: (&u32);
let mut _42: &u32;
let _43: &u32; // arg0
let mut _44: std::fmt::ArgumentV1<_>;
let mut _45: &u32;
let mut _46: for<r, 's, 't0> fn(&r u32, &s mut std::fmt::Formatter<t0>) -> std::result::Result<(), std::fmt::Error>;

```