arbitrary_enum_discriminant

The tracking issue for this feature is: #60553

The arbitrary_enum_discriminant feature permits tuple-like and struct-like enum variants with #[repr(<int-type>)] to have explicit discriminants.

Examples

```
#![feature(arbitrary_enum_discriminant)]

#[allow(dead_code)]
#[repr(u8)]
enum Enum {
    Unit = 3,
    Tuple(u16) = 2,
    Struct {
        a: u8,
        b: u16,
    } = 1,
}

impl Enum {
    fn tag(&self) -> u8 {
        unsafe { *(self as *const Self as *const u8) }
    }
}

assert_eq!(3, Enum::Unit.tag());
assert_eq!(2, Enum::Tuple(5).tag());
assert_eq!(1, Enum::Struct{a: 7, b: 11}.tag());
```