`intrinsics`

The tracking issue for this feature is: None.

Intrinsics are never intended to be stable directly, but intrinsics are often exported in some sort of stable manner. Prefer using the stable interfaces to the intrinsic directly when you can.

These are imported as if they were FFI functions, with the special `rust-intrinsic` ABI. For example, if one was in a freestanding context, but wished to be able to `transmute` between types, and perform efficient pointer arithmetic, one would import those functions via a declaration like

```
#![feature(intrinsics)]
# fn main() {}

extern "rust-intrinsic" {
    fn transmute<T, U>(x: T) -> U;

    fn offset<T>(dst: *const T, offset: isize) -> *const T;
}
```

As with any other FFI functions, these are always `unsafe` to call.