```
[[questions]]
id = "dd16401e-9f74-4c86-a6ec-da2937cb26e5"
type = "MultipleChoice"
prompt.prompt = """
Which call to this `find_until` function will cause a runtime panic?
```
fn find_until(mut v: Span<u64>, n: u64, til: usize) -> Option<usize> {
    let mut i = 0;
    loop {
        if i == til {
            break Option::None;
        }
        if *v[i] == n {
            break Option::Some(i);
        }
        i += 1;
    }
}
```
"""
prompt.distractors = [
  "`find_until(array![1, 2, 3].span(), 0, 0);`",
  "`find_until(array![1, 2, 3].span(), 3, 3);`",
  "`find_until(array![1, 2, 3].span(), 1, 4);`",
]
answer.answer = "`find_until(array![1, 2, 3].span(), 4, 4);`"
context = """
If `til = 4`, then for an array of length 3, the loop will attempt to index the array with `i =
3`,
which is out of bounds.  This function does not panic if `n = 1` because it returns before
reaching
the out-of-bounds index.
"""
[[questions]]
id = "2e6570eb-8bf4-48b7-9032-5815475bc412"
type = "Tracing"
prompt.program = """
fn main() {
    let mut v: Array<ByteArray> = array!["Hello "];
    let mut s = *v[0];
    s.append(@"world");
    println!("{s}");
}
"""
answer.doesCompile = false
answer.lineNumber = 3
```

context = """
As Cairo's memory layout is immutable, types cannot be moved out of an array by indexing. The only
possibility is to copy the value to a new variable using the `*` (desnap) operator. Therefore, as
ByteArray is not copyable, the line `let mut s = *v[0];` does not compile.
"""
[[questions]]
id = "95d528ee-ae78-4892-a438-a5d97f07f52a"
type = "Tracing"
prompt.program = """
fn main() {
    let mut v: Array<usize> = array![1, 2, 3];
    let mut i = *v[0];
    i += 1;
    println!("{}, {}", i, v[0]);
}
"""
answer.doesCompile = true
answer.stdout = "2, 1"
context = """
`*v[0]` copies the value of the first element of the array to `i`. Therefore, `i` is a copy of the
value of `v[0]`, and incrementing `i` does not affect `v[0]`.
"""