Exercise: Precedence and Associativity

Given the precedence table and associativity rules in the previous slides,

- Apply parentheses to the expression to show how operands are grouped to operators and
- Give the result of the expression
- Where a = 1, b = 2, c = 3, d = 2, e = 2, f = 3

Fortran a + b * c ** d ** e / f result is 55 (1 + ((2 * (3 ** (2 ** 2))) / 3)

Pascal a < b and c < d result is static semantic error because of (1 < (2 and 3)) < 2 --operands of logical and are not Boolean

C a < b && c < d result is false (1 < 2) && (3 < 2)

Exercise: Precedence, Associativity, Evaluation Order

Given the precedence table and associativity rules in the previous slides, and evaluation order within expression is left to right, what is the result of this C program?

```
int give2() { printf("two\n"); return 2; }
int give3() { printf("three\n"); return 3; }
int give4() { printf("four\n"); return 4; }

int main() {
  printf("%d\n", give4() + give2() * give3() - give4() / give2());
  return 0;
}
```

```
(give4() + (give2() * give3())) - (give4() / give2())
Result is
four
two
three
four
two
8
```

Exercise: Short-Circuit

How can we use short-circuit evaluation to make the following code safer?

```
const int MAX = 10;
int A[MAX];
…
if (A[i] > foo) …
```

Avoid out-of-bound subscripts:

if (((i >= 0) && (i < MAX)) && (A[i] > foo)) …

if (!(i < 0 || i >= MAX) && A[i] > foo) …

```
if (n/d < threshold) …
```

Avoid division by zero:

if ((d != 0) && (n/d < threshold)) …

Exercise: Case/Switch Implementation

What is the problem with jump table implementation in the previous slide?

Jump table implementation can consume large space.

If the set of labels includes large value range, the array of addresses will consume large space. Each value needs an entry in the array.

If the set of labels is not dense, the labels scatter and do not share arms, the list of arms may consume large space.

What do you think a compiler should do about it?

Use different implementations for different case statements. For example, if the number of labels is small, it can even use sequential testing as in nested if. If the value range is not large, it can use the jump table. Or it can use binary search instead.